AFRL-AFOSR-VA-TR-2016-0361

Motion Coordination and Adaptation Using Deception and Human Interactions

Magnus Egerstedt
GEORGIA TECH RESEARCH CORPORATION
505 10TH ST NW
ATLANTA, GA 30318-5775

11/18/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/RTA2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
06-10-2016	Final Report	01-01-2013 - 30-06-2016

4. TITLE AND SUBTITLE
Motion Coordination and Adaptation Using Deception and Human Interactions

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-13-1-0029

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Magnus Egerstedt and Panagiotis Tsiotras

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Georgia Institute of Technology
Atlanta, GA 30312

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AIR FORCE OFFICE OF SCIENTIFIC RES/AFOSR/ARLINGTON, VA

Air Force Office of Scientific Research
875 North Randolph Street
Suite 325, Room 3112
Arlington VA, 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release

DISTRIBUTION A: Distribution approved for public release.

13. SUPPLEMENTARY NOTES
Control Theory, Multi-Robot Teams

14. ABSTRACT

This project developed fundamental, new tools and techniques for how to
structure the coordination and control strategies in teams of mobile robots. In particular, two general thrust areas were pursued, focusing on human-swarm interactions and pursuit-evasion-based motion control strategies. Although interesting in their own rights, the unifying theme behind these two different thrusts is the notion of intent, where the first thrust, which can be thought of as evolving at a higher level of abstraction, focused on how user intent can be injected into a network of mobile agents in a fundamentally sound manner. The second thrust, in turn, focused on how the intent can be hidden in order to produce effective, deception-based coordination and pursuit strategies.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Magnus Egerstedt
					19b. TELEPHONE NUMBER *(Include area code)* (404) 894-3484

Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18
Adobe Professional 7.0

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Motion Coordination and Adaptation Using Deception and Human Interactions

FINAL REPORT, OCT. 2016

Magnus Egerstedt and Panagiotis Tsiotras

This project focuses on the development of fundamental new tools and techniques for how to structure the coordination and control strategies in teams of mobile robots. In particular, two general thrust areas are pursued, focusing on human-swarm interactions and pursuit-evasion-based motion control strategies. Although interesting in their own rights, the unifying theme behind these two different thrusts is the notion of *intent*, where the first thrust, which can be thought of as evolving at a higher level of abstraction, focuses on how user intent can be injected into a network of mobile agents in a fundamentally sound manner. The second thrust, in turn, focuses on how the intent can be hidden in order to produce effective, deception-based coordination and pursuit strategies.

1 Summary of Accomplishments

During the evolution of this project, a number of different results and directions have been pursued, and we here briefly summarize these and then discuss them in greater detail in subsequent sections.

a) How can effective interaction abstractions be established for teams of mobile robots that allow users to instantaneously effect the operation of the team? We developed a novel notion of *manipulability* for defining this abstraction. Manipulability was furthermore used to define *haptic* interactions that can be defined for enabling single operators to control and interact with teams of mobile robots. Since there is no unique nor canonical mapping from the swarm configuration to the forces experienced by the operator, multi-agent manipulability was selected as the candidate mapping, whereby the forces experienced by the operator relate to how directly input directions, injected at a leader agent in the network, translate to motions of the follower agents. Experimental results support the efficacy of the proposed, haptic, human-swarm interaction mapping, through user studies where operators are tasked with driving a collection of robots through a series of way points.

b) In order to establish what, ultimately, constitutes effective human-robot-team interactions, we need formal guarantees for whether or not a given human-swarm interaction (HSI) is appropriate for achieving multi-robot tasks. Examples of such tasks include guiding a swarm of robots into a particular geometric configuration. In doing so, we define what it means to impose a HSI structure on a multi-robot system. Several examples of multi-robot systems with unique HSI structures have been considered to illustrated the viability of the proposed approach.

c) One of the main challenges in human-swarm interactions is the construction of suitable abstractions that make an entire robot team amenable to human control. For such abstractions to be useful, they need to scale gracefully as the number of robots increases. In this work, we consider the use of time-varying density functions to externally influence a robot swarm. Density functions abstract away the size of the robot team and describe instead the concentration

of agents over the domain of interest. This allows a human operator to design densities so as to manipulate the robot swarm as a whole, instead of at the individual robot level. We have developed coverage of time-varying density functions as a mechanism to translate densities into robotic movement, and provided a series of control laws that guarantee optimal coverage by the robot team.

d) We considered the following differential game of pursuit and evasion involving two participating players: an evader, which has limited maneuverability, and an agile pursuer. The agents move on the Euclidean plane with different but constant speeds. Whereas the pursuer can change the orientation of its velocity vector arbitrarily fast, the evader cannot make turns having a radius smaller than a specified minimum turning radius. This problem can be seen as a reversed Homicidal Chauffeur game, hence the name "Suicidal Pedestrian Differential Game." We derived the optimal strategies of the two players and characterized the initial conditions that lead to capture if the pursuer acts optimally, and areas that guarantee evasion regardless of the pursuer's strategy. Both proximity-capture and point-capture were considered. After applying the optimal strategy for the evader, it was shown that the case of point-capture reduces to a special version of Zermelo's Navigation Problem (ZNP) for the pursuer. The ZNP is a classical problem of finding minimum-time optimal paths in the presence of flow fields. Therefore, the well-known ZNP solution can be used to validate the results obtained through the differential game framework, as well as to characterize the time-optimal trajectories.

   Having derived the optimal strategies of the two players and characterized the initial conditions that lead to capture if the pursuer acts optimally, and areas that guarantee evasion regardless of the pursuer's strategy, we then focused our efforts on its multi-player extension. Namely, utilizing the connection to Zermelo's Navigation Problem, we were able to generalize the two-player solution under some assumptions for the case of multiple pursuer UAVs.

e) We considered a relay pursuit-evasion problem with two pursuers and one evader, where only one of the pursuers is active in pursuit of the evader at each instant of time. We reduced the problem to a one-pursuer/one-evader problem subject to a state constraint and derived the optimal control strategy of the evader to maximize the capture time. We also proposed a suboptimal, yet practical, control strategy for the evader to prolong capture that does not require the solution of the corresponding two-point boundary-value optimal control problem. Extensions to the multiple-pursuer/one-evader case were also presented and evaluated via numerical simulations.

f) We investigated solutions of a pursuit-evasion game of two players in the presence of an external flow field. The external flow field is approximated by a time-invariant affine function. By utilizing standard techniques from differential game theory, we characterized the regions of initial conditions that lead to capture, as well as the regions that result in evasion when the two players act optimally. We derived the optimal strategies of the pursuer and the evader within the capture regions. We showed that in the presence of an external flow field, the player with an accurate knowledge of the field has an advantage. Finally, we presented numerical simulations of the resulting pursuer and evader trajectories for several values of the parameters of the external flow field.

g) We dealt with the problem of a team of pursuers distributed in the plane subject to an environmental disturbance (e.g., winds). The objective of the pursuers is to intercept a moving target which is not affected by the presence of the disturbance. We solved this problem by assigning only one pursuer to chase the target at every instant of time, based on a Voronoi-like partition of the plane. During the pursuit, the pursuer assignment changes dynamically based on this

partition. We presented an algorithm to efficiently update this Voronoi-like partition on-line and validated the theoretical results through numerical simulations.

h) We adopted a reachability-based approach to deal with the pursuit-evasion differential game between two players in the presence of dynamic environmental disturbances (e.g., winds, sea currents). We gave conditions for the game to terminate in terms of reachable set inclusions. Level set equations were defined and solved to generate the reachable sets of the pursuer and the evader. The corresponding time-optimal trajectories and optimal strategies can be retrieved immediately afterwards. We validated our method by applying it to a pursuit-evasion game in a simple flow field, for which an analytical solution is available. We also implemented the proposed scheme to a problem with a more realistic flow field.

i) We derived the min-max Differential Dynamic Programming (GT-DDP) algorithm for solving a large class of differential games in continuous time. We provided a set of backward differential equations for the value function expansion without assuming closeness of the initial nominal control to the optimal control solution, and derived the update law for the controls. We introduced the GT-DDP algorithm and analyzed the effect of the game theoretic formulation in the feed-forward and feedback parts of the control policies.

j) We developed a sampling-based algorithm designed to solve differential game and risk-sensitive stochastic optimal control problems. T he cornerstone of the proposed approach is the formulation of the problem in terms of forward and backward stochastic differential equations (FBSDEs). By means of a nonlinear version of the Feynman-Kac lemma, we obtained a probabilistic representation of the solution to the nonlinear Hamilton-Jacobi-Isaacs equation, expressed in the form of a decoupled system of FBSDEs. This system of FBSDEs can then be simulated by employing linear regression techniques. Utilizing the connection between stochastic differential games and risk-sensitive optimal control, we demonstrate that the proposed algorithm is also applicable to the latter class of problems. The algorithm was validated by simulations.

2 Human-Swarm Interactions

The emerging field of Human-Swarm Interactions (HSI) concerns itself with the problem of making large teams of autonomous agents amenable to human control. The rationale for this is the need to invert the current many-to-one relationship where multiple operators are required to control a single autonomous vehicle, such as an unmanned aerial vehicle. As many facets of society move towards greater levels of automation, this traditional mode of interaction is not sustainable. This becomes even more clear in the various branches of the military in general, and the air force in particular, where the trend clearly is pointing towards greater usage of autonomous vehicles coexisting with humans.

Not surprisingly, the issue of human control of unmanned vehicle fleets has received considerable attention during the last decade, mainly focusing on particular types of scenarios and on how the information flow should be structured so that the human operator is not overloaded yet retains sufficient situational awareness. What this project focused on was a fundamental treatment of the human-swarm interaction problem – not in terms of particular interfaces and algorithms, but in terms of system theoretic properties that human-swarm interaction structures should exhibit for the higher-level questions (interfaces, et.c.) to be meaningful beyond anecdotal and particular scenarios.

2.1 Haptic Manipulability

When controlling a team of robots via external inputs, it is desirable to know how the inputs are affecting the team's behavior. In some situations, it is desirable to have all of the agents moving as one collective unit, with each robot having the same velocity. Instead of broadcasting the signal to all robots, we control the movement of one robot directly and use local interaction laws to control the motion of the followers, as described in the previous section as leader- follower control. Manipulability describes how effectively the follower agents are being controlled by the leaders at any point in time. As opposed to controllability, which is a property that describes between which states the system can be in, manipulability is an instantaneous notion, and thus fits our needs for haptic feedback.

In order to define manipulability explicitly, we first need to introduce some notation. Consider a swarm of N mobile robots, consisting of N_f followers and N_l leaders, where $N_f + N_l = N$. At time t, robot i's position is given by $x_i(t) \in \mathbb{R}^d,\ i = 1, \ldots, N$, where d is the spatial dimension of the network, e.g., $d = 2$ corresponds to the case of planar robots, $d = 3$ represents agents that move in a three dimensional space, and so forth. The positions are aggregated together to give the overall position of the robot team at time t, $x(t) = [x_1^T(t), ..., x_N^T(t)]^T \in \mathbb{R}^{Nd}$. We assume that the indexing of the agents is such that the first N_f agents are the followers, and the last N_l agents are the leaders. Under this indexing scheme, $x(t) = [x_f^T(t), x_l^T(t)]^T$, where $x_f(t) = [x_1^T(t), ..., x_{N_f}^T(t)]^T \in \mathbb{R}^{N_f d}$ and $x_l(t) = [x_{N_f+1}^T(t), ..., x_N^T(t)]^T \in \mathbb{R}^{N_l d}$.

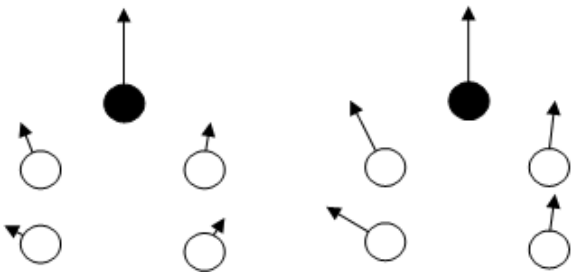
Now we can introduce the formal definition of manipulability, which is formulated as the ratio between the leaders' and the followers' velocities, i.e.,

$$M = \frac{\|\dot{x}_f\|^2}{\|\dot{x}_l\|^2}. \tag{1}$$

For example, if you have two multi-agent networks with the same velocity applied to the leader in each, the larger the magnitude of the followers' velocities, the higher the manipulability index. This is illustrated in Figure 1, where the robot configuration in 1(a) has a lower manipulability than that in 1(b). Here, since the leaders in each network have the same velocity, the difference in the follower velocities is due to a difference in interaction topology between the two networks. If the two networks had the same interaction topology, differences in manipulability would be caused by different leader velocities.

Since we want to use this manipulability index for haptic feedback, it is important to understand what all it depends on. It is clearly a function of $\dot{x}_l$, because that is the control input directly specified by the user. It also depends on where each of the robots are in space, x, as well as the structure of the multi-agent network, i.e., which robot pairs would like to maintain inter-robot distances. In this leader-follower network, the control law of the followers is designed so that adjacent agents maintain desired distances between each other. If we let $V = 1, ..., N$ denote the set of agents, then we can define the unordered set $E \subset V \times V$ to contain the robot pairs that are adjacent in the underlying network. By combining the *vertex* set V with the *edge* set E, we form the undirected graph $G = (V, E)$, which defines the information exchange network of the multi-agent team of robots. The manipulability index given in Equation (27) also depends on this graph.

We know that M in Equation (27) depends on $\dot{x}_l$, x, and G, but in order to compute it, we also



(a) Lower Manipulability

(b) Higher Manipulability

Figure 1: Manipulability comparison between two leader-follower multi-robot networks ($N_l = 1$). The filled circle in each network is the leader and the arrows represent the agents' velocities. Here, the network on the left has a lower manipulability than the network on the right due to the follower velocities being smaller in magnitude. In this case, the difference in the follower velocities between the two networks and thus the difference in manipulability would be caused by a difference in interaction topology (not shown).

need to know $\dot{x}_f$. However, the followers' velocities depend on the choice of interaction dynamics. This can be remedied by developing an approximate manipulability measure that does not depend on the interaction dynamics, i.e.,

$$\tilde{M} \approx \frac{\|\dot{x}_f\|^2}{\|\dot{x}_l\|^2}.$$

The approximate manipulability measure was derived by using the rigid-link approximation. The rigid-link approximation assumes that the desired distances $\{d_{ij}\}_{(v_i,v_j)\in E}$ between connected agents are perfectly maintained by the followers at all times. In other words, $||x_i(t) - x_j(t)|| = d_{ij}, \forall (v_i, v_j) \in E, t \geq 0$. Under the rigid link approximation, the distances between connected agents do not change over time. Given smooth and differentiable trajectories of $x_i(t)$, this means that

$$\frac{d}{dt}\|x_i(t) - x_j(t)\|^2 = 0, \ \forall (v_i, v_j) \in E, \ t \geq 0,$$

which expands to

$$(x_i - x_j)^T(\dot{x}_i - \dot{x}_j) = 0, \ \forall (v_i, v_j) \in E, \tag{2}$$

where, for the sake of notational simplicity, we have dropped the dependence on t.

Using (2), the rigid-link approximation condition can be written in matrix form as

$$R(x)\dot{x} = 0,$$

where $R(x) \in \mathbb{R}^{|E|\times Nd}$ is the so-called *rigidity matrix* of the system, and where $|E|$ is the cardinality of the edge set. Or, if we split this into the parts contributed by the leaders and the followers,

$$R(x, G)\begin{bmatrix} \dot{x}_f \\ \dot{x}_l \end{bmatrix} = [R_f(x, G)|R_l(x, G)]\begin{bmatrix} \dot{x}_f \\ \dot{x}_l \end{bmatrix} = 0,$$

where $R_f \in \mathbb{R}^{|E| \times N_f d}$ and $R_l \in \mathbb{R}^{|E| \times N_l d}$. This allows the follower velocities to be directly expressed as a function of the leader velocities, as

$$\dot{x}_f = -R_f^\dagger(x, G) R_l(x, G) \dot{x}_l, \tag{3}$$

where $R_f^\dagger$ is the Moore-Penrose pseudoinverse of R_f.

The approximate manipulability measure was derived using this relation in Equation (3) and is given by

$$\tilde{M}(x, \dot{x}_l, G) = \frac{\dot{x}_l^T J^T(x, G) J(x, G) \dot{x}_l}{\|\dot{x}_l\|^2}, \tag{4}$$

where $J(x, G) = -R_f^\dagger(x, G) R_l(x, G)$. This expression for approximate manipulability can now be used to provide a human operator with haptic feedback about the instantaneous effectiveness of his or her control inputs.

This approximate manipulability measure fits all of the criteria for a well-suited haptic force discussed in the previous section. First of all, when $N_l = 1$, it can be shown that

$$0 \leq \tilde{M} \leq N_f.$$

Because $\tilde{M}$ is the ratio of squared norms, it is clear that $\tilde{M} \geq 0$. As for the other side of the inequality, when $N_l = 1$, R_l can be expressed in terms of R_f as

$$R_l = -R_f \tilde{I}_f,$$

where $\tilde{I}_f = \mathbf{1}_{N_f} \otimes I_d$, where $\mathbf{1}_{N_f}$ is an N_f-dimensional column vector with 1s in all of its entries, $\otimes$ denotes the Kronecker product, and I_d denotes the $d \times d$ identity matrix. By substituting this R_l into Equation (3), we get

$$\dot{x}_f = -R_f^\dagger R_l \dot{x}_l = R_f^\dagger R_f (\mathbf{1}_{N_f} \otimes I_d) \dot{x}_l = R_f^\dagger R_f (\mathbf{1}_{N_f} \otimes \dot{x}_l).$$

Since $R_f^\dagger R_f$ is a projection matrix, we get $\|\dot{x}_f\|^2 \leq N_f \|\dot{x}_l\|^2$. Thus, the desired result,

$$\tilde{M} = \frac{\|\dot{x}_f\|^2}{\|\dot{x}_l\|^2} \leq N_f,$$

follows.

This means that approximate manipulability has a known minimum and maximum when there is one leader in the group. Since the case we are interested in is a single human operator controlling a single leader via a haptic device, this assumption is fine.

Manipulability describes how effectively the followers are being controlled by the leader, and a higher value of manipulability is better, so we want to use a haptic force that encourages the human operator to move the network of agents in directions that produce a high manipulability value. The question that remains is: how do we create a mapping from manipulability to haptic force? One option is to apply the force feedback in the direction opposite the one that the user is trying to move the leader in. In this case, the force would impede the motion of the user, so

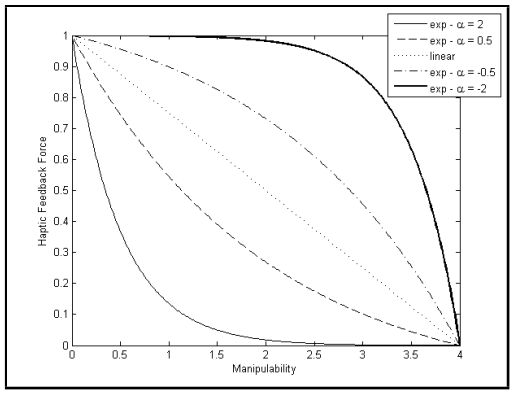


Figure 2: Manipulability vs haptic force mappings used in user experiments.

we map high manipulability to low force feedback and low manipulability to high force feedback. This way, if the user is moving the leader in a direction that produces a high manipulability of the swarm, little feedback force would resist the user's motion. Alternatively, if the user is moving the leader in a direction that produces low manipulability, the resistive force from the haptic device would encourage the user to choose a different direction in which to move the leader.

Using this type of force feedback, the approximate manipulability of the swarm and the haptic feedback are inversely related. There are many ways we can form an equation for the mapping. It seems reasonable that the maximum manipulability value should map to zero haptic feedback force, so the user does not experience any resistance when moving the leader in this desirable direction. Similarly, the minimum value of the approximate manipulability measure should map to the maximum force of the haptic device. Thus we need to map the range $[0, N_f]$ to the range $[0, H]$, where H is the maximum applicable force of the haptic device being used.

There are many ways that this mapping can be done and the simplest such way is through a linear mapping, given by

$$F_l(\tilde{M}) = H\left(1 - \frac{\tilde{M}}{N_f}\right), \tag{5}$$

where F_l is the magnitude of the haptic force meant to resist the user's motion, $\tilde{M}$ is the approximate manipulability of the team, and N_f is the number of followers in the swarm.

This linear map does not encourage high values of manipulability in a particularly powerful way. Another option is an inverse exponential map, given by

$$F_e(\tilde{M}) = H\,\frac{e^{-\alpha\tilde{M}} - e^{-\alpha N_f}}{1 - e^{-\alpha N_f}}. \tag{6}$$

Here, α is a parameter that can be changed to adjust the rate of change of the force as a function of manipulability. Figure 2 contains a plot showing the linear mapping and the exponential mapping for α values of -2, -0.5, 0.5, and 2. The plot was made using $N_f = 4$ and $H = 1$. It can be seen that as α becomes more negative, the resistive feedback force remains high for a larger range of manipulability values, indicating that the mapping should be more forceful in encouraging high values of manipulability.

Now that we know that approximate manipulability is an appropriate team-level property for use

Figure 3: Initial setup of the robot team, where the leader agent is the right-most robot. The two target locations are circled.

Figure 4: Depicted is a student utilizing haptic device while looking at the virtual environment (middle screen).

with haptic feedback, we will describe the experiments used as proof of concept for this method. In fact, we implemented haptic manipulability feedback as described previously using a haptic device known as the PHANTOM Omni by SensAble Technologies. The force feedback was rendered and applied to the device while a user controlled a swarm of robots under the leader-follower configuration. The user controlled the velocity of the single leader robot with the haptic device at the same time that the manipulability forces were being fed back through the device.

A network of five mobile agents was created using Khepera III differential drive robots. The user was instructed to move the group of robots to one target location and then to the other, in either order, using the haptic device to control the leader robot. Each target location was marked with an 'X' on the floor. The initial setup of the robots, along with the target locations, can be seen in Figure 3. In addition to the physical robots, a virtual environment was provided to the user to show the robots' locations, along with the target locations. Completion was defined by the leader being on top of the target locations. Figure 4 shows one of the students using the haptic device to control the swarm while looking at the virtual environment.

The network configuration chosen for this team of robots was a triangle formation with two extra robots on the ends, as shown in Figure 5. The lines in the diagram represent the edges, or the links that represent which agents are maintaining distances with each other. Here, if the agents are numbered 1 through 5 from left to right, agent 1 is only connected to agent 2 and agent 5 is only connected to agent 4. Therefore, the network is not rigid, meaning it does not necessarily stay

in the same shape as shown in Figure 5 while it is being moved by the user.

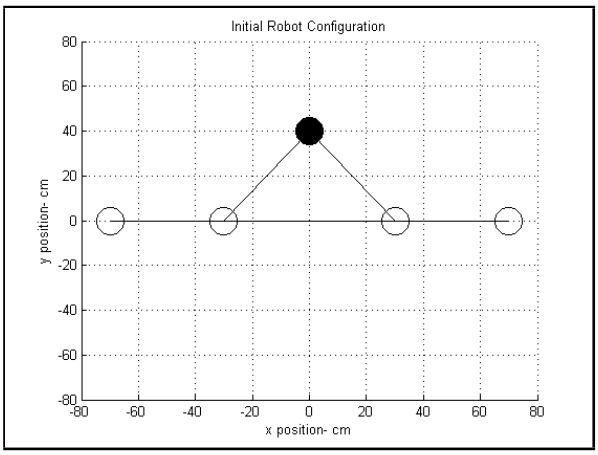


Figure 5: Initial configuration of multi-robot team, showing the connections between agents. The filled circle represents the leader.

The haptic force magnitude given by F_e and F_l in the previous section, after being computed using the instantaneous approximate manipulability of the system, must be converted into a value that can be applied to the haptic device. Only the x and z degrees of freedom of the haptic device were used for this experiment. When the subject used the device to control the velocity the leader, the x and z positions are captured by the program and converted into a velocity by computing the norm and the direction angle. Let x_{pos} and z_{pos} be the positions obtained from the haptic device. The magnitude and angle are computed as in Equations (7) and (8).

$$mag = \sqrt{z_{pos}^2 + x_{pos}^2} \tag{7}$$

$$\theta = \arctan(\frac{z_{pos}}{x_{pos}}) \tag{8}$$

After the magnitude of the haptic feedback force is computed, which is a value between 0 and H, it is used to scale the input velocity given by the user. Here we take H to be 1. This is slightly different than the forces mentioned in the previous section. This is done so that the magnitude of the feedback force generated by the haptic device can be at most the magnitude of the force exerted by the user onto the device. In other words, the haptic feedback force will only be as strong as the force exerted by the user onto the device while he or she is controlling the motion of the leader. In practice, this allows the force feedback to come on gradually and helps keep the haptic device from becoming unstable.

In addition, both the x and z components are negated so that the output force is in the opposite direction of the input velocity. Thus, as described previously, the force that the haptic device generates is a repulsive force that is intended to impede the motion of the haptic device in certain directions. Below are the equations used for the force rendering, where F represents the function that maps the approximate manipulability, $\tilde{M}$, of the swarm to the haptic force (either F_l or F_e from the previous section), and mag and θ come from Equations (7) and (8), respectively. The x and z haptic forces that are published to the PHANTOM Omni after being computed are given below as F_x and F_z, while the y component is always set to zero.

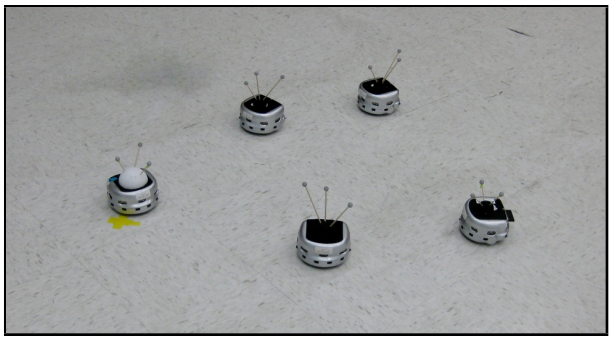


Figure 6: Depicted is a team of Khepera III robots nearing a target location.

$$F_x = -\operatorname{sgn}(x_{pos})\, F(\tilde{M})\, mag\, |\cos(\theta)| \tag{9}$$

$$F_z = -\operatorname{sgn}(z_{pos})\, F(\tilde{M})\, mag\, |\sin(\theta)| \tag{10}$$

The haptic feedback described here was successfully implemented and the user was able to move the swarm of robots while feeling the resistive forces of the haptic device. The team of robots moving during one of the experiments can be seen in Figure 6, where the leader is the robot with the white styrofoam object on top of it and the 'X' on the floor marks one of the target locations.

2.2 A Formal Approach to Human-Swarm Interactions

Our objective is to determine whether it is possible for a human operator to use a particular human-swarm interaction (HSI) to achieve some geometric configuration with a swarm of mobile robots. To establish feasibility, we first need to know what a HSI represents in terms of the structure it imposes on a multi-robot system, and what it means for a human operator to achieve a particular geometric configuration with the robotic swarm.

In general, we consider continuous-time, time-invariant systems with inputs, which represent robotic swarms that can be externally controlled (or interacted with) by a user. The dynamics of such multi-robot systems can be defined as $\dot{x}(t) = f(x(t), u(t))$, where $x(t) \in \mathcal{X}$ is the state of the system at time t and $u(t) \in \mathcal{U}$ is the input to the system at time t. In fact, $x(t)$ will represent the stacked vector of the states belonging to all robots at time t, while $x_i(t)$ will refer to the state of robot i at time t. For example, $x(t)$ will typically represent the position or pose of all robots together at time t.

More importantly, the differentiable function $f : \mathcal{X} \times \mathcal{U} \to T\mathcal{X}$, where $T\mathcal{X}$ is a tangent space, is structured according to the network topology of the multi-robot system. The network topology is given by a graph $\mathcal{G} = (V, E)$, where V is the set of vertices representing the agents, and E is the set of edges representing information exchange between agents via communication links or due to sensor footprints. Specifically, $f \in \text{sparse}_{\mathcal{X}}(\mathcal{G})$ conveys that state information in the multi-robot system can only flow between agents that are linked in the network topology. Consequently,

$$f \in \text{sparse}_{\mathcal{X}}(\mathcal{G}) \Leftrightarrow \left(j \notin \overline{N}(i) \Rightarrow \frac{\partial f(x,u)}{\partial x_i} = 0,\ \forall x, u \right), \tag{11}$$

where $N(i)$ is the so-called neighborhood of robot i, i.e., $j \in N(i)$ if $(i,j) \in E, i,j \in V$, and $\overline{N}(i) = N(i) \cup \{i\}$.

By picking a particular HSI structure, we are being specific about the structure of $\mathcal{U}$, i.e., how the user can interact with the robotic swarm. Our definition is as follows:

A HSI structure is a map

$$H : \mathcal{X} \times \mathcal{V} \to \mathcal{U}, \tag{12}$$

where $\mathcal{V}$ is some set of admissible inputs to make the corresponding robotic swarm more amenable to human control. Additionally,

$$f(x, H(x,v)) = f_H(x,v) \in sparse_{\mathcal{X}}(\mathcal{G}), \tag{13}$$

which means that the dynamics f under this map H needs to observe the sparsity structure imposed by the network topology.

This definition of a HSI structure implies that the control input to the system is really a combination of state feedback and a restricted set of inputs from the user, which respects the constraints imposed by the network topology. Consequently, the dynamics of a multi-robot system under such a HSI structure are

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ &= f(x(t), H(x(t), v(t)) \\ &= f_H(x(t), v(t)). \end{aligned} \tag{14}$$

Therefore, a HSI structures is a very specific way in which the user controls the multi-robot system, i.e., interacts with the robotic swarm.

For example, suppose that a robotic swarm consists of n mobile robots positioned on a rail ($x_i(t) \in \mathbf{R}$) with single-integrator dynamics,

$$\dot{x}_i(t) = u_i(t), \ i = \{1, \ldots, n\}, \tag{15}$$

where the control input for the first $n-1$ robots is

$$u_i(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)). \tag{16}$$

$N(i)$ denotes the neighborhood of robot i, which is the set of all its immediate neighbors in the network topology derived from communication links or sensor footprints.

The control input for the n-th robot is

$$u_n(t) = v(t), \ v(t) \in \mathcal{V}, \tag{17}$$

which corresponds to the user directly controlling the position of the n-th robot. This HSI structure is commonly referred to as a *single-leader network*, because the user interacts with the swarm of robots by guiding a "leader" robot, while the other robots follow the leader and each other according to the consensus dynamics in (16).

If we stack all $x_i(t)$'s into a state vector $x(t) \in \mathbf{R}^n$ and all $u_i(t)$'s into an input vector $u(t) \in \mathbf{R}^n$, then the ensemble dynamics of our example system are

$$\dot{x}(t) = u(t) = -L_f x(t) + lv(t),\ l = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbf{R}^n \tag{18}$$

where L_f is a sub-block of the graph Laplacian L with all entries in the n-th row of L equal to zero. Consequently, the single-leader network HSI structure is a particular structuring of the control input $u(t)$ in (18) given by the function H, such that

$$u(t) = H(x(t), v(t)) = -L_f x(t) + lv(t), \tag{19}$$

where $v(t) \in \mathbf{R}$ is the user input.

Finally, when a multi-robot system under some HSI structure can asymptotically converge to a state, a subset of states, or all states in a specification set $\mathcal{S}$ and stay in this set, then

$$\limsup_{t \to \infty} d(x(t), \mathcal{S}) = 0, \tag{20}$$

where,

$$d(x(t), \mathcal{S}) = \inf_{s \in \mathcal{S}} \|x(t) - s\|. \tag{21}$$

If this is true, then we say that the user can *achieve* some or all of the geometric configurations described by $\mathcal{S}$ with the robotic swarm.

These definitions of HSI structures, together with stability guarantees using Control Lyapunov Functions and measures of user attention and effort, were investigated on a number of different HSI structures, including broadcast signals, leader-follower control, and manipulation of geometric swarm aggregates.

2.3 Human-Swarm Interactions via Coverage

Coverage control is one area of multi-agent control that has received significant attention lately and it is concerned with how to position agents in such a way that "surveillance" of a domain of interest is maximized. In this context, an idea that has been widely adopted to describe how

interesting a "domain of interest" is, is to associate a density function to the domain.However, the focus of previous coverage algorithms has largely been on static density functions, which does not provide enough flexibility when human operators are to adaptively interact with a team through a dynamic re-shaping of the density functions. This research is, in contrast, able to handle time-varying density functions as human inputs to the swarm, and Fig. 7 illustrates how the proposed approaches allow a user to manipulate a team of robots to drive to certain areas of the environment.

The Coverage Problem

We are looking for for decentralized update laws that allow a human operator to influence the swarm of agents by providing a density function characterizing the areas of importance within the domain. The agents in the robot swarm will seek to optimize their coverage over the domain under the influence of the user-provided density. In order to discuss optimal coverage, one first has to associate a cost to a robot configuration that describes how well a given area is being covered.

Let $D \subset \mathbb{R}^2$ be the two-dimensional convex domain representing the robot workspace. Moreover, let $\phi : D \times [0, \infty) \to (0, \infty)$ be the associated density function, which we will assume is bounded and continuously differentiable in both arguments, and where $\phi(q, t)$ captures the relative importance of a point $q \in D$ at time t. For the sake of Human-Swarm Interactions (HSI), this density function will be an user-provided input to the robot swarm.

The coverage problem involves placing n robots in D, and we let $p_i \in D$, $i = 1, \ldots, n$ be the position of the i^{th} robot. Moreover, the domain itself is divided into regions of dominance,

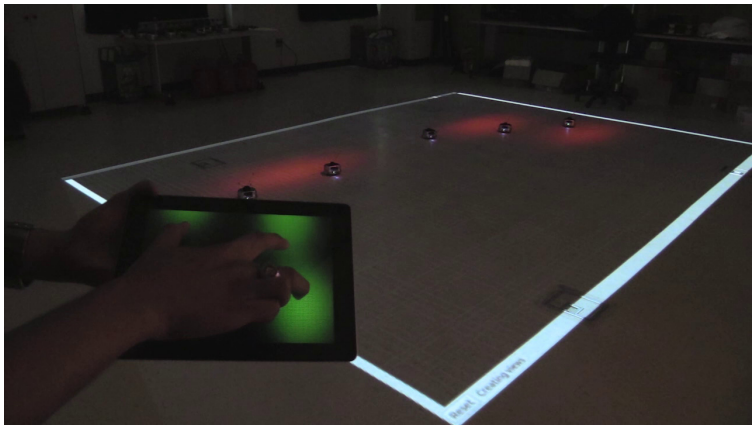


Figure 7: A team of five Khepera robots are influenced by a human operator to drive to certain parts of the environment. The tablet is representative of the robot environment. An overhead projector allows us to overlay the broadcasted density on the environment in real-time.

$P_1, \ldots, P_n$ (forming a proper partition of D), where the idea is to let robot i be in charge of covering region P_i. One can then ask how good the choice of p and P is, where $p = [p_1^T, \ldots, p_n^T]^T$, and $P = \{P_1, \cdots, P_n\}$. The final piece needed to answer this question is a measure of how well a given point $q \in D$ is covered by robot i at position $p_i \in D$.

We consider the locational cost given by

$$\mathcal{H}(p, P, t) = \sum_{i=1}^{n} \int_{P_i} \|q - p_i\|^2 \phi(q, t) dq. \tag{22}$$

At a given time t, when a configuration of robots p together with the partition P minimize, the domain is said to be optimally covered with respect to ϕ. However, it is possible to view the minimization problem as a function of p alone, by observing that given p, the choices of P_i that minimize (22) is

$$V_i(p) = \{q \in D \mid \|q - p_i\| \leq \|q - p_j\|, i \neq j\}.$$

This partition of D is a Voronoi tessellation — hence the use of V_i to denote the region. With this choice of region, we can remove the partition as a decision variable and instead focus on the locational cost

$$\mathcal{H}(p, t) = \sum_{i=1}^{n} \int_{V_i(p)} \|q - p_i\|^2 \phi(q, t) dq \tag{23}$$

where $\phi(q, t)$ is the user-provided input to the system.

It can be shown that

$$\frac{\partial \mathcal{H}}{\partial p_i} = \int_{V_i} -2(q - p_i)^T \phi(q, t) dq, \tag{24}$$

and since $\phi > 0$, one can define the mass m_i and center of mass c_i of the i^{th} Voronoi cell, V_i, as

$$m_i(p, t) = \int_{V_i(p)} \phi(q, t) dq, \tag{25}$$

$$c_i(p, t) = \frac{\int_{V_i(p)} q\phi(q, t) dq}{m_i}. \tag{26}$$

For convenience, we will stack the masses into a single diagonal (positive-definite) matrix denoted by

$$M = \begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_n \end{bmatrix} \otimes I_d, \tag{27}$$

where "$\otimes$" is the Kronecker product and I_d is the $d \times d$ identity matrix where d corresponds to the dimension of the domain ($d = 2$ is considered here). We also group the centers of mass as a single

vector $c = \left[c_1^T, \ldots, c_n^T\right]^T$. Using these quantities allows us to rewrite the partial derivative in (24) as

$$\frac{\partial \mathcal{H}}{\partial p_i} = 2m_i(p_i - c_i)^T. \tag{28}$$

From this expression, we can see that a critical point of (47) is

$$p_i(t) = c_i(p, t), \quad i = 1, \cdots, n, \tag{29}$$

and a minimizer to (47) is necessarily of this form. Moreover, when (29) is satisfied, p is a so-called centroidal Voronoi tessellation (CVT).

Centralized Coverage of Time-Varying Densities

We are now in pursuit of a control laws that allow us achieve a CVT, even if the density function is time-varying. In order to move forward with our analysis, we need to understand some of the properties associated with the center of mass of a given Voronoi tessellation. We first note that the partial derivatives of the center of mass with respect to the agents positions has eigenvalues bounded by one, as long as the agents are close enough to a CVT, and under the assumption that at the CVT the locational cost is locally strongly convex.

Lemma 1. *If the CVT is locally strongly convex, then there exists $\delta > 0$ such that if*

$\|p(t) - c(p(t), t)\| < \delta$ *then* $\mathrm{Re}(\lambda) < 1$ *for all* $\lambda \in \mathrm{eig}(\frac{\partial c}{\partial p})$.

Corollary 1. *As long as the agents are close enough to a CVT configuration for which the locational cost is locally strongly convex, the matrix inverse $(I - \frac{\partial c}{\partial p})^{-1}$ exists.*

We use this corollary to obtain our first result on maintaining a CVT for all time.

Theorem 1 (Maintaining CVT)**.** *Let $p\left(t_0\right) = c\left(p\left(t_0\right), t_0\right)$. If*

$$\dot{p} = \left(I - \frac{\partial c}{\partial p}\right)^{-1} \frac{\partial c}{\partial t}, \qquad t \geq t_0$$

then

$$\left\|p\left(t\right) - c\left(p\left(t\right), t\right)\right\| = 0, \qquad t \geq t_0$$

as long as the inverse $\left(I - \partial c/\partial p\right)^{-1}$ is well-defined.

Proof: Assume the agents begin from a CVT configuration, i.e., $p\left(t_0\right) = c\left(p\left(t_0\right), t_0\right)$. We need to ensure that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(p\left(t\right) - c\left(p\left(t\right), t\right)\right) = 0, \qquad \forall t \geq t_0.$$

But this implies that $\dot{p} = \dot{c} = \frac{\partial c}{\partial p}\dot{p} + \frac{\partial c}{\partial t}$, which can be rearranged into the form

$$\dot{p} = \left(I - \tfrac{\partial c}{\partial p}\right)^{-1} \tfrac{\partial c}{\partial t}$$

as long as the inverse is well-defined. From Lemma 1, a sufficient condition for this is that the initial CVT makes the locational cost locally strongly convex. □

In order for this theorem to be meaningful, we require that the agents first achieve a CVT configuration which is then maintained. This control law also assumes the agents will be able to instantaneously accelerate to the required velocities, which may not be possible in practice. In order compensate for saturation, modeling errors and deviations from the CVT, a proportional term can be introduced that forces the agents into a CVT. This update law is called the *TVD-C* which stands for time-varying densities, centralized case.

$$\dot{p}(t) = \left(I - \frac{\partial c}{\partial p}\right)^{-1} \left(-\kappa \left(p(t) - c(p(t), t)\right) + \frac{\partial c}{\partial t}\right) \tag{30}$$

where $\kappa > 0$ is a proportional gain. It is noteworthy that this proportional term influences the team of robots to move towards a (scaled) gradient descent direction to achieve a CVT configuration, and that once a CVT is achieved the proportional term does not contribute to the update law and Theorem 1 holds.

As long as the inverse is well-defined, this update law will drive the agents into a CVT. This is encapsulated in the following theorem.

Theorem 2 (Convergence of TVD-C)**.** *If* $1 \notin \operatorname{eig}\left(\frac{\partial c}{\partial p}\right)$ *and we let*

$$\dot{p} = \left(I - \frac{\partial c}{\partial p}\right)^{-1} \left(-\kappa \left(p - c\right) + \frac{\partial c}{\partial t}\right), \qquad t \geq t_0$$

then $\|p(t) - c(p(t), t)\| \to 0$ *as* $t \to \infty$ *with rate of decrease* $\exp\left(-\kappa(t - t_0)\right)$.

Proof:
Consider the total derivative for $\|p(t) - c(p(t), t)\|^2$ below

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\|p - c\|^2\right) &= 2\left(p - c\right)^T \left(\dot{p} - \dot{c}\right) = 2\left(p - c\right)^T \left(\dot{p} - \frac{\partial c}{\partial p}\dot{p} - \frac{\partial c}{\partial t}\right) \\
&= 2\left(p - c\right)^T \left(\left(I - \frac{\partial c}{\partial p}\right)\dot{p} - \frac{\partial c}{\partial t}\right) \\
&= 2\left(p - c\right)^T \left(\left(I - \frac{\partial c}{\partial p}\right)\left(I - \frac{\partial c}{\partial p}\right)^{-1} \left(-\kappa\left(p - c\right) + \frac{\partial c}{\partial t}\right) - \frac{\partial c}{\partial t}\right) \\
&= -2\kappa \|p - c\|^2
\end{aligned}$$

which tells us that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\|p-c\|^2\right) = -2\kappa\|p-c\|^2$$

This is a homogeneous linear differential equation which for initial condition $\|p(t_0) - c(p(t_0), t_0)\|$ has solution given by

$$\begin{aligned} & \|p(t)-c(p(t),t)\|^2 = \exp(-2\kappa(t-t_0))\|p(t_0)-c(p(t_0),t_0)\|^2 \\ \Rightarrow \quad & \|p(t)-c(p(t),t)\| \ = \exp(-\kappa(t-t_0))\|p(t_0)-c(p(t_0),t_0)\|. \end{aligned}$$

Hence $\|p(t) - c(p(t), t)\| \to 0$ as $t \to \infty$. Further, we see that

$$\begin{aligned} & \|p(t)-c(p(t),t)\| = \exp(-\kappa(t-t_0))\|p(t_0)-c(p(t_0),t_0)\| \\ \Rightarrow \quad & \frac{\|p(t)-c(p(t),t)\|}{\|p(t_0)-c(p(t_0),t_0)\|} = \exp(-\kappa(t-t_0)) \end{aligned}$$

such that the rate of decrease is given by $\exp(-\kappa(t - t_0))$. □

Distributed Coverage of Time-Varying Densities

Recall that by combining equations (25) and (26), we have that

$$c_i(p,t) = \frac{\int_{V_i(p)} q\,\phi(q,t)\ \mathrm{d}q}{\int_{V_i(p)} \phi(q,t)\ \mathrm{d}q},$$

which depends on p in the boundary of the area over which the two integrals are taken.

In order to compute these partials, we first need to make use of Leibniz rule.

Lemma 2. *Let $\Omega(p)$ be a region that is a smooth function of p such that the unit outward normal vector n is uniquely defined almost everywhere on $\partial\Omega$, which is the boundary of Ω. Let*

$$F = \int_{\Omega(p)} f(q)dq.$$

Then

$$\frac{\partial F}{\partial p} = \int_{\partial\Omega(p)} f(q)\frac{\partial q}{\partial p}\cdot n(q)dq$$

where $\frac{\partial q}{\partial p}$ is the derivative of the points on $\partial\Omega$ with respect to p.

It can now be shown that for any point $q \in \partial V_{ij}$ (the boundary between adjacent cells V_i and V_j),

$$\frac{\partial q}{\partial p_j^{(b)}} \cdot (p_j - p_i) = \frac{1}{2} e_b \cdot (p_j - p_i) - e_b \cdot \left(q - \frac{p_i + p_j}{2} \right),$$
$$\frac{\partial q}{\partial p_i^{(b)}} \cdot (p_j - p_i) = \frac{1}{2} e_b \cdot (p_j - p_i) + e_b \cdot \left(q - \frac{p_i + p_j}{2} \right),$$

where $p_j^{(b)}$ denotes the b^{th} component of the vector p_j and e_b is the b^{th} elementary unit vector. Note that in this chapter, $b = 1, 2$ since we are considering the case $D \subset \mathbb{R}^2$ only.

Substituting this into Leibniz rule, we obtain

$$\frac{\partial c_i^{(a)}}{\partial p_j^{(b)}} = -\left(\int_{\partial V_{ij}} \phi q^{(a)} \frac{q^{(b)} - p_j^{(b)}}{\|p_j - p_i\|} dq \right) \Big/ m_i + \left(\int_{V_i(P)} \phi q^{(a)} dq \right) \left(\int_{\partial V_{ij}} \phi \frac{q^{(b)} - p_j^{(b)}}{\|p_j - p_i\|} dq \right) \Big/ m_i^2 \quad (31)$$

where $a = 1, 2$, $b = 1, 2$ and where $i \neq j$. When $i = j$ we must consider the contribution from all neighbors

$$\frac{\partial c_i^{(a)}}{\partial p_i^{(b)}} = \sum_{k \in N_{V_i}} \left[\left(\int_{\partial V_{ik}} \phi q^{(a)} \frac{q^{(b)} - p_i^{(b)}}{\|p_k - p_i\|} dq \right) \Big/ m_i - \left(\int_{V_i(P)} \phi q^{(a)} dq \right) \left(\int_{\partial V_{ik}} \phi \frac{q^{(b)} - p_i^{(b)}}{\|p_k - p_i\|} dq \right) \Big/ m_i^2 \right]. \quad (32)$$

We can rewrite these equations more compactly using block matrix notation

$$\left[\frac{\partial c}{\partial p} \right]_{ij} = \frac{\partial c_i}{\partial p_j} = -\frac{\int_{\partial V_{ij}} (q - c_i)(q - p_j)^T \phi \, \mathrm{d}q}{m_i \|p_j - p_i\|}, \quad (33)$$

$$\left[\frac{\partial c}{\partial p} \right]_{ii} = \frac{\partial c_i}{\partial p_i} = \sum_{k \in N_{V_i}} \frac{\int_{\partial V_{ik}} (q - c_i)(q - p_i)^T \phi \, \mathrm{d}q}{m_i \|p_j - p_k\|}. \quad (34)$$

where $[\cdot]_{ij}$ corresponds to the i^{th}, j^{th} $d \times d$ block matrix.

It is noteworthy that given a continuously differentiable density function ϕ, computing $\partial c / \partial p$ at any given time t becomes an exercise in finding line and area integrals. In implementation, it

suffices to use numerical approximations to compute these integrals (e.g., Riemann sums, Gaussian quadrature rule).

One more partial derivative is required for update law (30), namely $\partial c/\partial t$. Another application of Leibniz rule results in

$$\frac{\partial c_i}{\partial t} = \frac{m_i \int_{V_i} q \frac{\partial \phi}{\partial t}(q,t)\,\mathrm{d}q - \int_{V_i} q\phi(q,t)\,\mathrm{d}q \int_{V_i(p)} \frac{\partial \phi}{\partial t}(q,t)\,\mathrm{d}q}{m_i^2} \tag{35}$$

or more compactly

$$\frac{\partial c_i}{\partial t} = \frac{\int_{V_i}(q-c_i)\frac{\partial \phi}{\partial t}\,\mathrm{d}q}{m_i}. \tag{36}$$

with $\frac{\partial c}{\partial t}^T = [\frac{\partial c_1}{\partial t}^T, \ldots, \frac{\partial c_n}{\partial t}^T]$.

When implementing the update law, it suffices to numerically compute the integrals in $\partial c/\partial t$. However, unlike with the computation of the integrals in $\partial c/\partial p$, we require knowledge of $\partial \phi/\partial t$. If ϕ is not provided analytically in t, then one could:

1. Utilize a finite difference scheme to approximate $\partial \phi/\partial t$. This could however give rise to difficulties with noisy measurements.

2. Alternatively, the user could provide the time evolution of the density function directly via a continuous function $\partial \phi/\partial t$ such that $\phi(q,t) = \int_{t_0}^{t} \frac{\partial \phi}{\partial t}(q,\tau)\,\mathrm{d}\tau$. In the implementations, the "shape" of the density is defined as a continuously differentiable function $\phi(q,t_0)$ defined over D, and the as the user provided input corresponds to $\partial \phi/\partial t$.

As a final implementation note, finding the integrals over V_i may be computationally intensive. However, the expanded versions of these partial derivatives reveal that only a few integrals actually need to be computed per agent. To compute all expressions it suffices to compute the integrals of ϕ, $d\phi/dt$, $q\phi$, and $qd\phi/dt$ over V_i, which may be computed once for each agent (and may be computed in a distributed fashion).

The second subtle difficulty with update law (30) is ensuring that the inverse $(I - \partial c/\partial p)^{-1}$ is well defined, which is in general hard to do. Lemma 1 provides us with a sufficient condition for the existence of this matrix and the desired approximation can be found by using the Neumann series.

Lemma 3 (Neumann series)**.** *Let A be a square matrix. If $\lim_{k\to\infty} A^k = 0$, then $I - A$ is invertible and*

$$(I-A)^{-1} = I + A + A^2 + A^3 + \ldots.$$

Moreover, for a $m \times m$ square matrix A, $\lim_{k\to\infty} A^k = 0$ if and only if $|\lambda_i| < 1$ for all $i = 1, 2, \cdots, m$, where λ_i are the eigenvalues of A. As such, let λ_{max} denote the eigenvalue with the largest magnitude of the matrix $\partial c/\partial p$. Using the Neumann series, we can express $(I - \partial c/\partial p)^{-1}$ as

$$\left(I - \frac{\partial c}{\partial p}\right)^{-1} = I + \frac{\partial c}{\partial p} + \left(\frac{\partial c}{\partial p}\right)^2 + \dots \tag{37}$$

as long as $|\lambda_{max}| < 1$.

Our goal will be to truncate this series to obtain the well-defined approximation to the matrix inverse, but then the question arises: how many terms should be kept? The answer lies in the sparsity structure of $\partial c/\partial p$.

Given a Voronoi partition of the area of interest, we denote the boundary between the two cells V_i and V_j by ∂V_{ij}. Since we are only considering the planar case, there are three possibilities for ∂V_{ij}:

1. ∂V_{ij} is empty, meaning that cells V_i and V_j do not intersect.
2. ∂V_{ij} consist of a single point, meaning that cells V_i and V_j share a single vertex.
3. ∂V_{ij} is a line, meaning that cells V_i and V_j share a face.

We will denote N_{V_i} to be set of indexes pertaining to the agents whose Voronoi cells V_j share a face with agent i's Voronoi cell V_i.

Lemma 4. $j \notin N_{V_i} \implies \dfrac{\partial c_i}{\partial p_j} = 0.$

Proof: For the first two cases, i.e., ∂V_{ij} is either empty or consists of a singleton, from (31) and (32) we see that the integrals over ∂V_{ij} would be zero. Note that for these two cases, this will be true for all four elements in $\partial c_i/\partial p_j$. Since these two cases correspond to agents i and j not sharing a face, we conclude that $j \notin N_{V_i}$ implies that $\partial c_i/\partial p_j = 0$.

This lemma tells us that $\partial c/\partial p$ actually encodes adjacency information of the graph induced by the Voronoi tessellation. This induced graph is known as the *Delaunay graph*.

To obtain a distributed update law, we must insist that the update for $\dot{p}_i$ depends only on information from itself (p_i and $\phi(q,t)$ for $q \in V_i$) and information on neighboring agents (p_j and $\phi(q,t)$ for $q \in V_j$, for all $j \in N_{V_i}$). To this end, we truncate the Neumann series in (37) after just two entries, i.e., $\left(I - \partial c/\partial p\right)^{-1} \approx I + \partial c/\partial p$. By modifying update law (30) with this approximation,

we obtain the update law

$$\dot{p} = \left(I + \frac{\partial c}{\partial p}\right)\left(-\kappa\left(p - c\right) + \frac{\partial c}{\partial t}\right),$$

which at the individual robot level results in the update law called *TVD-D*$_1$ for Time-Varying Densities, Distributed case with 1-hop adjacency information:

$$\dot{p}_i = \frac{\partial c_i}{\partial t} - \kappa\left(p_i - c_i\right) + \sum_{j \in \overline{N}_{V_i}} \frac{\partial c_i}{\partial p_j}\left(\frac{\partial c_j}{\partial t} - \kappa\left(p_j - c_j\right)\right) \tag{38}$$

where $\overline{N}_{V_i}$ is the closed neighborhood set to Voronoi cell i in the Delaunay graph.

It should be noted that (38) is always well-defined (as long as ϕ is continuously differentiable). In other words, even if the Neumann series is not convergent or if the inverse does not exist, the entries in (38) are well-defined. In fact, it turns out that during the robotic experiment, even in cases where $|\lambda_{max}| > 1$, the robots consistently evolve in a manner that achieves coverage.

One can now investigate what happens when higher order terms are kept in the Neumann series. For this, we let $dist(i, j)$ denote the edge distance, or number of edges in the shortest path, between i and j in the Delaunay graph induced by the Voronoi tessellation. And, as $\partial c/\partial p$ is a (block) adjacency matrix, we have that

$$\left[(\partial c/\partial p)^k\right]_{ij} \neq 0 \;\Rightarrow\; dist(i,j) = k,\; k = 0, 1, 2, \ldots$$

where $[\ \cdot\ ]_{ij}$ denotes the block corresponding to cell c_i and robot position p_j.

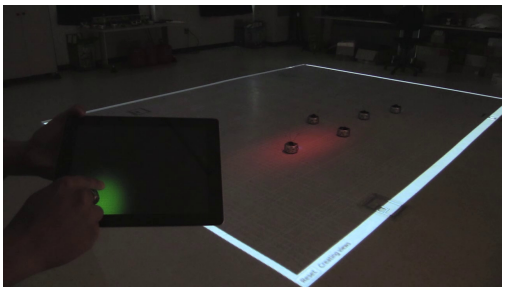
The k-hop version of *TVD-D*$_1$ thus becomes

$$\dot{p} = \sum_{\ell=0}^{k} \left(\frac{\partial c}{\partial p}\right)^{\ell}\left(-\kappa(p - c) + \frac{\partial c}{\partial t}\right). \tag{39}$$

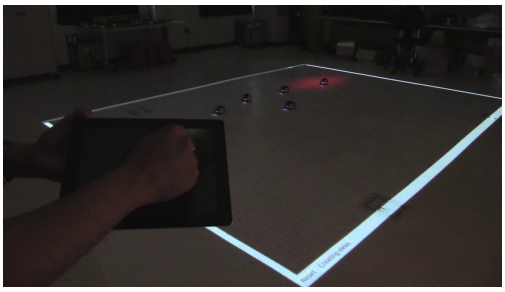
Robotic Experiments

These controllers for human-swarm interaction where implemented on robotic platforms to validate the approaches to HSI. For every agent, only it and its neighbors' state and density information in their respective Voronoi cells were used to compute the update laws. Line integrals were approximated by Riemann sums, whereas Gaussian quadrature rule was used for area integrals. The human operator used an iPad to input the density locations. These were transmitted over WiFi and UDP to a central computer which calculated the update law for each agent. Even though a central computer is used for the computation, the control is computed for every agent only using adjacency information, and then it is transmitted to the pertinent robot via WiFi and UDP.

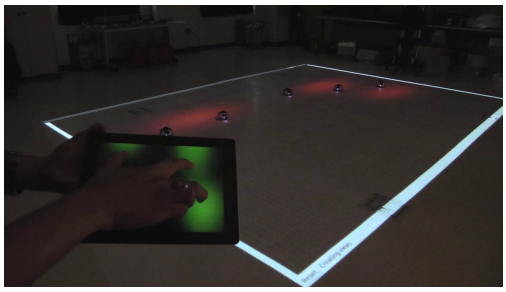
The density design was implemented on an Ubuntu computer with an Intel dual core CPU 2.13GHz and 4GB of memory, running ROS (Robot Operating System, version Diamondback). This computer also received state information from ten OptiTrack S250e motion capture cameras that were used to provide position and orientation data for the state information of an agent and its neighbors was used to compute the Voronoi tessellation. The robotic platforms used for the experiments were Khepera III robots from K-team. The Khepera III robots each have a 600MHz ARM processor with 128Mb RAM, embedded Linux, differential drive wheeled robots equipped with a wireless card for communication over a wireless router. The rviz package in ROS was used for visualization of the user-provided density function. The visualization was overlapped with the real physical environment to give a real-time visual representation of the user-provided density function. This is shown in Fig. 8.



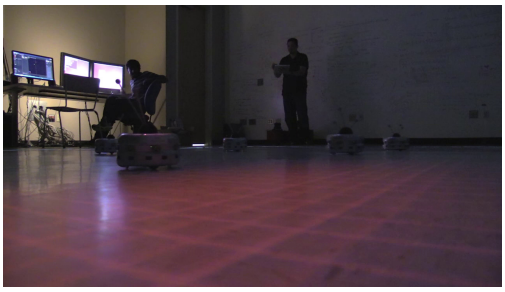
(a) User influencing swarm to center of domain



(b) User influencing swarm to corner of domain



(c) User splits the swarm by introducing multiple locations of interest



(d) User observes position of swarm and redirects it with the use of a tablet

Figure 8: Robotic implementation of HSI via coverage of Gaussian functions. A tablet is used to directly influence the location of the swarm formation. An overhead projector is used to visualize the user-provided density function in real-time over the robot workspace.

3 Pursuit-Evasion Problems

The main motivation behind this work has been our desire to extend differential game and pursuit-evasion problems to the case of multiple agents (both pursuers and evaders) while, at the same time, take into consideration the effect of external known or unknown disturbances (for example, wind fields in the case of unmanned aerial vehicles or sea currents for the case of marine vehicles). External disturbances, in particular, open the possibility of some of the agents (evaders or pursuers) using additional information about the environment in order to achieve their objective faster or more efficiently. Such concealment or deceptive strategies lead to games of asymmetric information that have been studied extensively in economic game theory but not so much in the context of pursuit-evasion games.

The overall objective is thus to increase the ambiguity in terms of the team intentions as they proceed to complete their common mission objective. As a particular instance of the proposed methodology we have investigated the problem of group pursuit of a maneuvering target by a team of unmanned vehicles. Since it is *not* be assumed a priori that the target is slower than its pursuers (as is typically the case for the majority of pursuit-evasion games of this type) coordination amongst the pursuing agents is necessary to ensure capture. The problem of group pursuit of a maneuvering target deals with the characterization of the individual pursuit strategies for each agent from the group of pursuers such that the collective objective of capturing the target is achieved. Capture of the target can either take place by only one pursuer (the pursuer that will reach the target first), or by multiple pursuers simultaneously. The group pursuit problem becomes more challenging if one seeks for pursuit strategies that require only local information (distributed group pursuit problem) and/or when the target is moving faster than the pursuers.

The overarching objective of this research is therefore to study motion coordination problems for a group of adversarial agents, whose aim is to accomplish certain collective objectives of the group while maintaining, at all times, their actions concealed from an adversarial opponent.

The benefits of this research to the US military are significant, since they will allow the deployment of teams of autonomous agents (e.g., UAVs) with increased decision-making capabilities. The results from this research can increase the scope and types of missions these teams of autonomous agents will be able to undertake. The collaborative strategies developed under the proposed research plan can ensure higher mission success for multi-agent systems, at a time when all branches of the military assign an ever increasing, more active, role to autonomous systems in the battlefield. Taking advantage of the external disturbances (as opposed to, say, try to cancel or nullify them) can yield better time and/or fuel-optimal solutions, while the possibility of asymmetric knowledge of the external disturbances (along with the dilemma of revealing this additional knowledge or not to the opponent) can lead to deceptive strategies that outperform strategies derived based on the common, full-information assumption.

3.1 The Suicidal Pedestrian Differential Game

In this research topic we assumed a pursuit-evasion game where the pursuer is agile but faster than the evader, but the evader is subject to maneuverability constraints. This problem is the reverse of the celebrated homicidal chauffeur game of Isaacs. Our objective was to find the optimal strategies of both the evader and the pursuer and also find the boundary of the winning set of the pursuer, that is, all initial conditions of the pursuer that will lead to capture. Both point capture and proximity capture were investigated.

The outcome of this game is defined by the game parameters (speed ratio and maneuverability restrictions on the evader) and the initial conditions (initial relative distance between the players and the evader's initial velocity vector orientation). The objective was to describe the relation between this set of parameters/initial conditions and the game outcome.

Problem Formulation and Main Results.

The equations of motion for the pursuer and the evader, written in an inertial frame of reference with coordinates x and y are given by

$$\dot{x}_p = v_p \cos\phi_p, \tag{40}$$
$$\dot{y}_p = v_p \sin\phi_p, \tag{41}$$
$$\dot{x}_e = v_e \cos\phi_e, \tag{42}$$
$$\dot{y}_e = v_e \sin\phi_e, \tag{43}$$
$$\dot{\phi}_e = -\frac{v_e}{R}u, \qquad u \in [-1, 1], \tag{44}$$

To proceed, we transform the problem from previous fifth-dimensional *realistic game space* to a two-dimensional *reduced game space*, by fixing the origin of a rotating coordinate frame at E's current position and by aligning the y-axis with E's velocity vector. The evader action then consists of choosing her center of curvature at a point $C = (R/u, 0)$ on the x-axis. Consequently, the reduced game space has only two coordinates, namely the (x, y) coordinates of P relative to E in the evader's fixed, velocity-aligned frame. The equations of motion of P in this frame are given by

$$\dot{x} = -\frac{v_e}{R}yu - v_p \sin\phi, \tag{45}$$
$$\dot{y} = \frac{v_e}{R}xu - v_e - v_p \cos\phi, \qquad u \in [-1, 1], \tag{46}$$

where ϕ is the pursuer's relative heading in this new reference frame, given by $\phi \triangleq \pi - \phi_p + \phi_e$. The game terminates when capture occurs, that is, when the relative distance between the evader and the pursuer becomes less than ℓ. The manifold contained within the game space which, once penetrated,

signals the game termination, is called the *terminal surface*. The terminal surface for our game is a circle with radius ℓ centered at the origin, i.e., the evader's position. We may thus formally define the terminal surface by $\mathcal{C} \triangleq \{\mathbf{x} \in \mathbb{R}^2 : |\mathbf{x}| = \ell\}$ and the game space as $\mathcal{E} \triangleq \{\mathbf{x} \in \mathbb{R}^2 : |\mathbf{x}| \geq \ell\}$. The terminal surface is then the boundary of the game space.

Retaining the concept of the payoff as it appears in the theory of zero-sum games, we assign the values $+1$ for escape (termination does not occur) and -1 for capture (termination occurs, i.e., the circle is penetrated by the pursuer's trajectory). An additional step is necessary in the process of constructing the barrier of the game. Namely, we need to distinguish the critical case in which the state $\mathbf{x}$ reaches the terminal surface but does *not* cross it, ultimately returning back into the interior of $\mathcal{E}$. The case when the terminal surface is reached without penetration, is referred to as *neutral* outcome, and the corresponding payoff is assigned zero value. Thus, given an initial state $\mathbf{x}(t_0)$ at $t = t_0$ and the pursuer and evader control histories, $\phi(\cdot)$ and $u(\cdot)$ respectively, the payoff formally reads as

$$J(\mathbf{x}(t_0), \phi(\cdot), u(\cdot)) = \begin{cases} +1, & \text{for no termination (escape),} \\ 0, & \text{neutral outcome,} \\ -1, & \text{for termination (capture).} \end{cases} \tag{47}$$

We thus seek to solve the problem of conflicting actions represented by u (maximizing control) and ϕ (minimizing control) that maximize/minimize the payoff (47) under the dynamic equations (45) and (46). Our goal is to obtain an analytic expression for the barrier surface, which consists of all starting points that lead to the neutral outcome.

The first step towards a solution of the game is to obtain the *usable part* of the terminal surface $\mathcal{C}$. It is not uncommon for a terminal surface of a game to be divided into two regions: the *Usable Part (UP)* and the *Nonuseable Part*, which are separated by what is known in the literature as the *Boundary of the Usable Part (BUP)*. The usable part is the subset of the terminal surface on which the pursuer can enforce capture, namely, penetration of the terminal surface. The nonusable part is the remaining part of the terminal surface. On it the game would end only if the evader does not play optimally.

To obtain the UP, we parameterize the terminal surface with the variable s as shown in Figure 9, by

$$(x, y) = (\ell \sin s, \ell \cos s), \qquad \mathbf{x} \in \mathcal{C}, \tag{48}$$

and, omitting several details, the usable part of the terminal surface turns out to satisfy

$$\cos s > -\frac{v_p}{v_e}, \qquad s \in (\frac{\pi}{2}, \pi], \tag{49}$$

while the BUP is thus determined through

$$\bar{s} = \arccos(-\frac{v_p}{v_e}), \qquad \bar{s} \in (\frac{\pi}{2}, \pi]. \tag{50}$$

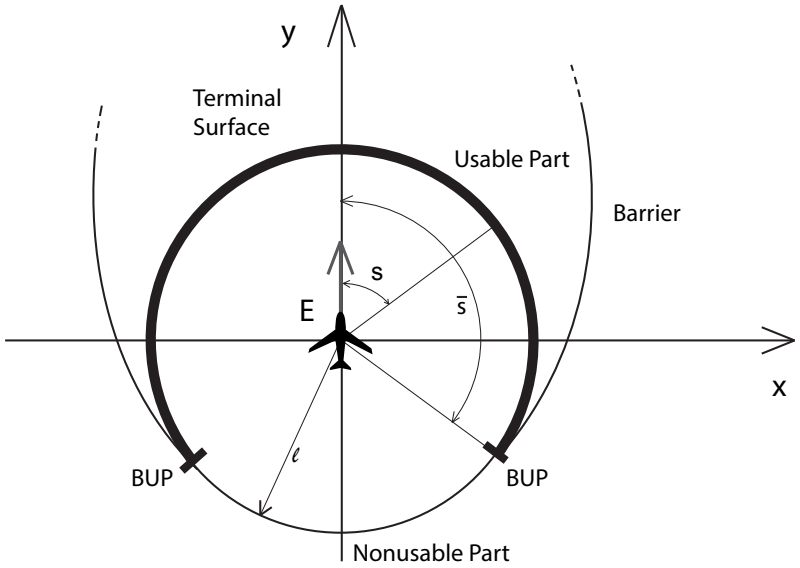


Figure 9: The terminal surface $\mathcal{C}$ of the game, which is a circle of radius ℓ. It is divided into the usable part (the dark line), and the nonusable part, separated by the BUP. The BUP connects to the barrier, which meets the terminal surface at the BUP tangentially.

An illustration of the usable part, the BUP and the nonusable part is given in Figure 9.

Having identified the BUP, we turn our attention to the construction of the barrier. The barrier is a *semipermeable surface*, that is, optimal play by both agents starting from any point will generate a trajectory that does not penetrate this surface. Without getting much into details, we enforce the Isaacs' equation for games of kind, obtain the retrogressive path equations and after applying the suitable boundary conditions, we connect the barrier tangentially to the BUP. The analytic expression of the barrier curve turns out to be:

$$x(\tau) = -R + R\cos(c\tau) + (\ell + v_p\tau)\sin(\bar{s} - c\tau), \tag{51}$$

$$y(\tau) = R\sin(c\tau) + (\ell + v_p\tau)\cos(\bar{s} - c\tau), \qquad \tau \in [0, \tau_{\max}], \tag{52}$$

where τ is the retrograde time $(t_f - t)$, and $\bar{s}$ is given by (50). Equations (51) and (52) separate the game space into two regions; a region in which optimal play of the pursuer leads to capture and a region in which optimal play of the evader leads to evasion. To obtain $\tau_{\max}$, it is important to note that the barrier expression is invalidated as soon as two barrier branches intersect – the part of the barrier arc beyond the point of intersection is then no longer valid and is therefore discarded. In our case, the two branches of the barrier intersect on the y-axis, because of the inherent symmetry of the problem at hand. Thus, we may obtain $\tau_{\max}$ as the root of $x(\tau) = 0$, i.e., $\tau_{\max}$ is the solution

of the transcendental equation

$$(\ell + v_p \tau_{\max}) \sin(\bar{s} - c\tau_{\max}) = R - R\cos(c\tau_{\max}). \tag{53}$$

Figure 10 depicts the barrier for $v_e = 1$, $v_p = 0.6$, $R = 0.7$, $\ell = 0.5$. Notice that the barrier meets

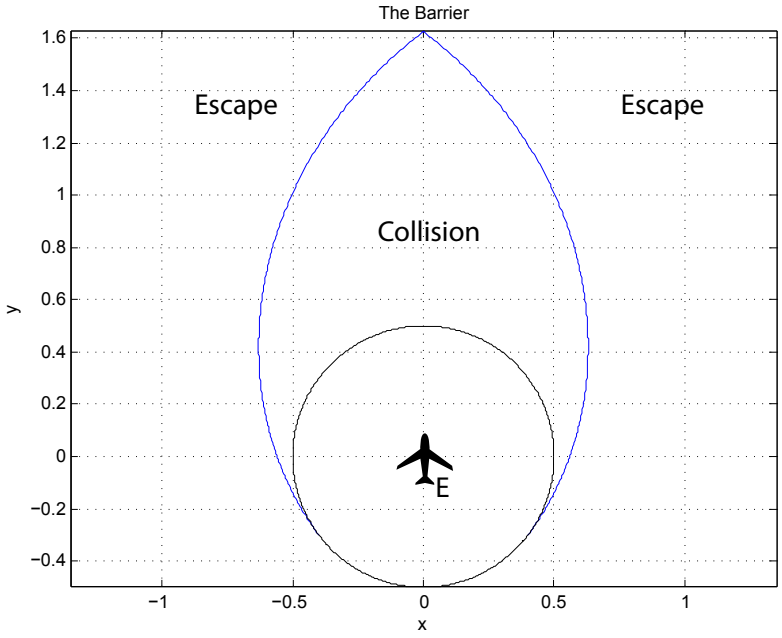


Figure 10: The barrier, given by equations (51) and (52), for $v_e = 1$, $v_p = 0.6$, $R = 0.7$, $\ell = 0.5$. Notice that the barrier meets the terminal surface tangentially at the BUP.

the terminal surface at the BUP tangentially.

Remark 1. *The cases in which the agents have the same speed, or point capture is considered, can be obtained by directly substituting* $v_p = v_e$ *or* $\ell = 0$ *respectively.*

After answering the question of which states leads to capture as opposed to evasion, we investigated the time-optimal problem when the outcome is capture, that is, we considered initial states within the capture zone delineated by the barrier and examine the characteristics of the time-optimal capture trajectories. We demonstrate that, for the special case of point-capture, the time optimal problem is equivalent to Zermelo's Navigation Problem in optimal control.

Point Capture: Time-Optimal Trajectories and Connection to Zermelo's Navigation Problem

Zermelo's Navigation Problem (ZNP) is a well-known result in optimal navigation, which has received a lot of attention in the literature. Initially stated by the German mathematician E. Zermelo in 1931, the problem formally reads: *"In a given vector field of currents, which is a function of position (and possibly time), a vehicle moves with constant speed relative to the currents. How should the vehicle be navigated in order to reach a given destination in minimum time?"*. In ZNP, the equations of motion of the vehicle are given by

$$\dot{x} = v\cos\phi + U(x,y), \tag{54}$$
$$\dot{y} = v\sin\phi + Q(x,y), \tag{55}$$

where U, Q are known functions that correspond to the components of the vector field along the x and y directions, respectively, and ϕ is the heading angle with respect to the x-axis (the control input). The goal is to minimize time until the vehicle reaches a target location.

Returning to the original differential equations (45)-(46), it is easy to observe that since E's optimal strategy is $u = -1$ (for initial conditions of the pursuer in the right-half plane), equations (45) and (46) assume the same form, with a change in the angle convention. Thus, E's optimal control results in an induced vector field akin to a current, which P needs to overcome in order to intercept E in minimum time. This vector field is shown in Figure 11. This fact allows us to use the well-known *Zermelo's Navigation Formula* which states that the optimal control ϕ^* obeys

$$\dot{\phi}^* = \sin^2\phi^*\frac{\partial Q(x,y)}{\partial x} + \sin\phi^*\cos\phi^*\left(\frac{\partial U(x,y)}{\partial x} - \frac{\partial Q(x,y)}{\partial y}\right) - \cos^2\phi^*\frac{\partial U(x,y)}{\partial y}, \tag{56}$$

which, for $U(x,y) = v_e y/R$ and $Q(x,y) = -v_e x/R - v_e$, yields

$$\dot{\phi}^* = \frac{v_e}{R}. \tag{57}$$

The problem therefore reduces to a two-point boundary value problem consisting of integrating the state equations along with (57), subject to initial conditions (x,y) and $\phi^*(0)$ that will lead to a trajectory passing through the origin $(0,0)$. Alternatively, one can consider integrating this system of ODEs backwards in time, i.e., by flipping the sign of the ODE's and using the variable τ, subject to the retrograde boundary conditions $(x,y) = (0,0)$ and a variable retrograde boundary condition $\phi_f^* \in [0,\ 2\pi]$ for (57). This will yield a parametric family of curves, and it remains to locate the one that passes through the original point (x,y) of interest. In fact, this integration can be performed analytically to obtain the following parametric family of curves

$$x(\phi_f^*;\tau) = -R + R\cos(c\tau) + v_p\tau\sin(\phi_f^* - c\tau), \tag{58}$$
$$y(\phi_f^*;\tau) = R\sin(c\tau) + v_p\tau\cos(\phi_f^* - c\tau), \qquad \tau \in [0,\tau_{\max}], \tag{59}$$

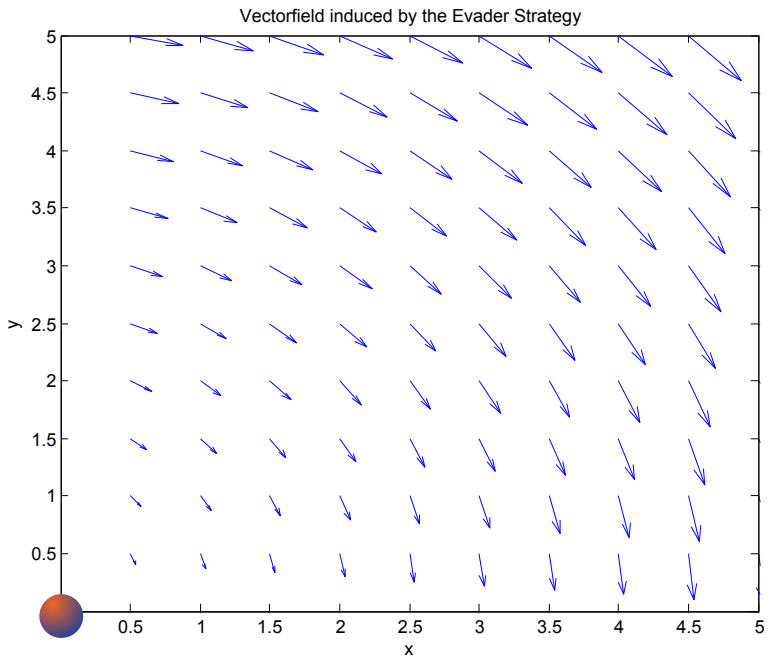


Figure 11: Applying the evader's strategy induces a vector field that resembles a current, which P needs to overcome in order to intercept E in minimum time. Plotted for $v_p = v_e = 1$, $R = 0.7$.

where ϕ_f^* is the free parameter and $\tau_{\max}$ is the solution to the transcendental equation

$$v_p \tau_{\max} \sin(\phi_f^* - c\tau_{\max}) = R - R\cos(c\tau_{\max}). \tag{60}$$

Figure 12 illustrates several time optimal trajectories, members of the parametric family of curves given by (58) and (59), corresponding to different values of ϕ_f^*. The barrier, i.e., the rightmost time optimal trajectory, is obtained for $\phi_f^* = \pi$, and is identical to the barrier.

Domain Decomposition in the Case of a Faster Pursuer

In this section we investigate the case when the pursuer is not only completely agile, but is also faster than the evader (that is, the speed ratio $\alpha = v_p/v_e > 1$). As already stated, this case leads to global capturability, namely, the evader is captured regardless of the initial conditions of the pursuer and the evader. Although capture is guaranteed on the entire Euclidean plane, it is shown that the Euclidean plane can be partitioned by a boundary $\mathcal{B}$ separating two different types of solutions exhibited in the game: one in which the evader is captured while performing a hard turn, and another one in which the evader, having completed the turn, is captured during a straight-line dash while trying to avoid the pursuer (also referred to as the *end game*). The region inside the

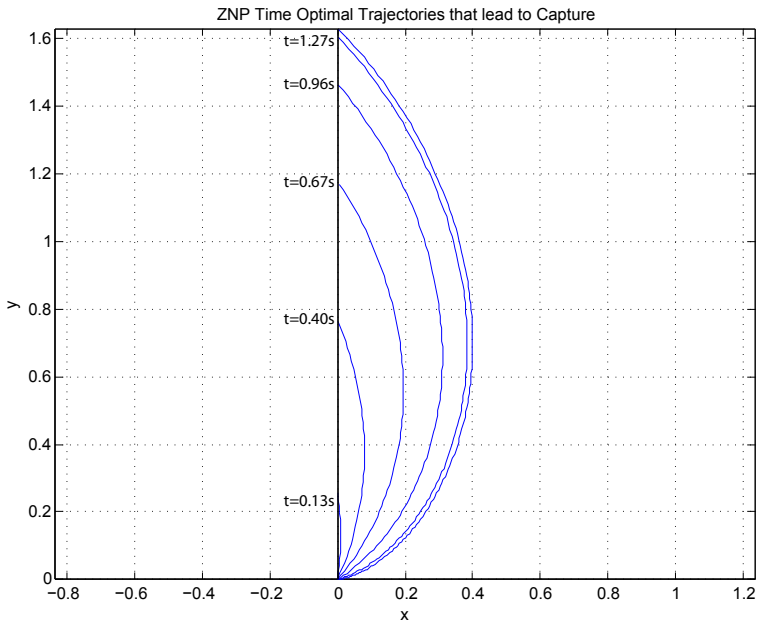


Figure 12: Members of the parametric family of curves given by (58) and (59), corresponding to different values of ϕ_f^*, $v = 1$, $R = 0.7$. The time to capture is also shown for the case in which the game is initialized as a head-on engagement. For clarity, only the case $u^* = -1$ is shown. The curves for $u^* = 1$ are symmetric with respect to the y-axis.

boundary $\mathcal{B}$ corresponds to initial conditions that lead to the first type of solution. For initial conditions of the game outside of the boundary, the second type of solution manifests itself. The shape of the boundary $\mathcal{B}$ resembles in appearance the one of the barrier for the case $\alpha \leq 1$. A geometric procedure to obtain the boundary $\mathcal{B}$ in the *realistic plane* leads, in the case of point capture, to a boundary defined by

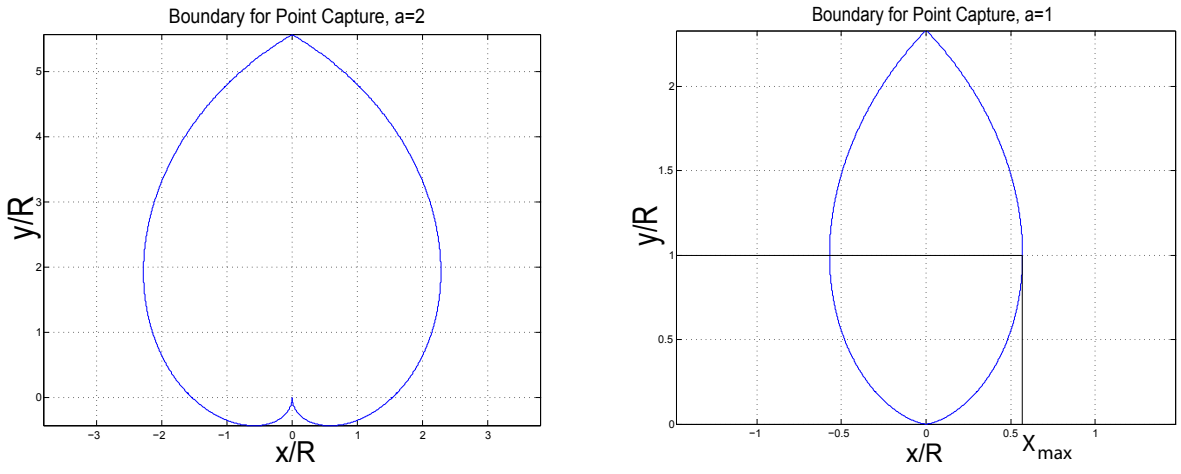
$$x(\theta)/R = \cos\theta + \alpha\theta\sin\theta - 1, \tag{61}$$

$$y(\theta)/R = \sin\theta - \alpha\theta\cos\theta. \tag{62}$$

The boundary in this case is depicted in Figure 13

Extension to the Case of Multiple UAVs

We turned our attention to the case in which the evader faces several pursuers. To this end, consider the game involving one evader and an arbitrary number – say, N– of pursuers, moving on



(a) (b)

Figure 13: The boundary $\mathcal{B}$ in the case of point capture, for (a) $\alpha = 2$, and (b) $\alpha = 1$. For $\alpha > 1$ the area enclosed by these curves corresponds to initial conditions that lead to capture while the evader is turning, whereas initial conditions outside the area enclosed by the curve lead to capture after the evader has completed her turn and while she performs a straight-line dash away from the evader (end game). For $\alpha = 1$ this curve separates the regions of capture (inside) and escape (outside) and it thus coincides with the barrier.

a plane. The subscripts p_i and e will be reserved to designate the "i-th pursuer" and the "evader," respectively. As before, the pursuers' objective is *capture*, that is, interception of the evader in finite time, whereas the evader's objective is *evasion*, a state in which she avoids interception indefinitely. Here, interception is to be understood in the context of coincidence of the Cartesian coordinates between the evader and at least one of the pursuers (point-capture). This will make the subsequent analysis a bit easier, but the results obtained can be modified to account for proximity capture as well. For this scenario, and for the sake of simplicity, it will be assumed that all agents have the same constant speed, that is, $v_e = v_{p_i} = v$ for all $i = 1, \dots, N$. As before, the pursuers are assumed to be agile, in the sense that they can change the direction of their velocity vector instantaneously. On the other hand, the evader is less agile and cannot make turns that have a radius smaller than a minimum turning radius R. The dynamic equations consist of N copies of the suicidal pedestrian dynamics, i.e.,

$$\dot{x}_i = -\frac{v}{R} y_i u - v \sin \phi_i, \tag{63}$$

$$\dot{y}_i = \frac{v}{R} x_i u - v \cos \phi_i - v, \qquad i = 1, \dots, N, \quad u \in [-1, 1], \tag{64}$$

where $(x_i, y_i) \in \mathbb{R}^2$ are the coordinates of the i-th pursuer in the reference frame fixed on the evader position, aligned with its velocity vector, and ϕ_i is the i-th pursuer's control, given by $\phi_i \triangleq \pi - \phi_{p_i} + \phi_e$. We will now make the following assumption:

Assumption 1. *For all $i = 1, \ldots, N$, we have $y_i(0) > 0$.*

This assumption essentially translates into all the pursuers populating at $t = 0$ only the upper half plane; in other words, the evader faces all the pursuers at the beginning of the game, rather than having pursuers at her back (or, alternatively, that all UAV's at her back are sufficiently far away to be neglected). Recall that, if pursuers are located both in front of the evader as well as at her back, and the evader is in the interior of the convex hull generated by the pursuer positions, then a completely agile evader will have no strategy leading to evasion. Since the set of evader strategies in our problem is just a subset of the set of strategies of a completely agile evader, capture is guaranteed in our case as well. Furthermore, it is reasonable to assume that once a head-on collision has been avoided by the evader against all incoming UAVs, these will not turn around to pursue the evader.

Game Termination

We will now establish the conditions under which the game terminates. Game termination occurs if the position of at least one of the pursuers becomes identical with that of the evader (collision), or if all pursuers enter the lower half plane (evasion). The latter condition guarantees evasion because no velocity vector component of the evader points towards a pursuer, so the evader is able to escape without changing her heading. This condition for evasion is equivalent to the evader being able to construct a hyperplane separating her from the pursuers and orient her velocity vector perpendicular to that hyperplane, as depicted in Figure 14. In our game setup, since the evader velocity vector is at all times aligned with the y-axis, we may take that hyperplane to be the x-axis, arriving at the same conclusion.

Extending the Barrier of the Suicidal Pedestrian Game

When the two agents have equal speed capabilities, the expression for the barrier is

$$x(\tau) = -R + R\cos(c\tau) + v\tau\sin(c\tau), \tag{65}$$

$$y(\tau) = R\sin(c\tau) - v\tau\cos(c\tau), \qquad \tau \in [0, \tau_{\max}], \tag{66}$$

where $\tau = t_f - t$ is the retrograde time variable starting from game termination, say at t_f, and $c = v/R$. To obtain $\tau_{\max}$, recall that the barrier expression in the two player game is invalidated as soon as two barrier branches intersect – the part of the barrier arc beyond the point of intersection is then no longer valid and is therefore discarded. If the evader is able to terminate the game without capture, then the evader is able to induce a trajectory that leads to the lower half plane in the evader fixed reference frame. This is equivalent to the evader being able to successfully eliminate her velocity vector component pointing towards the pursuer before capture occurs. The

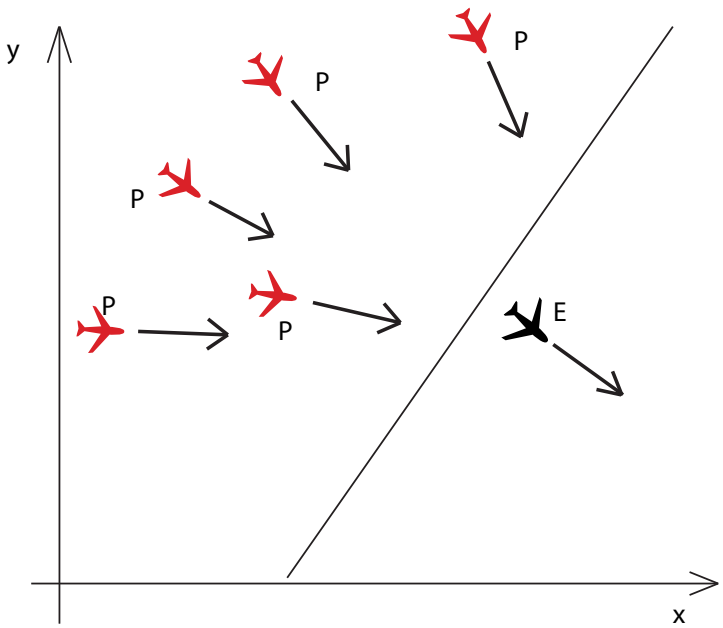
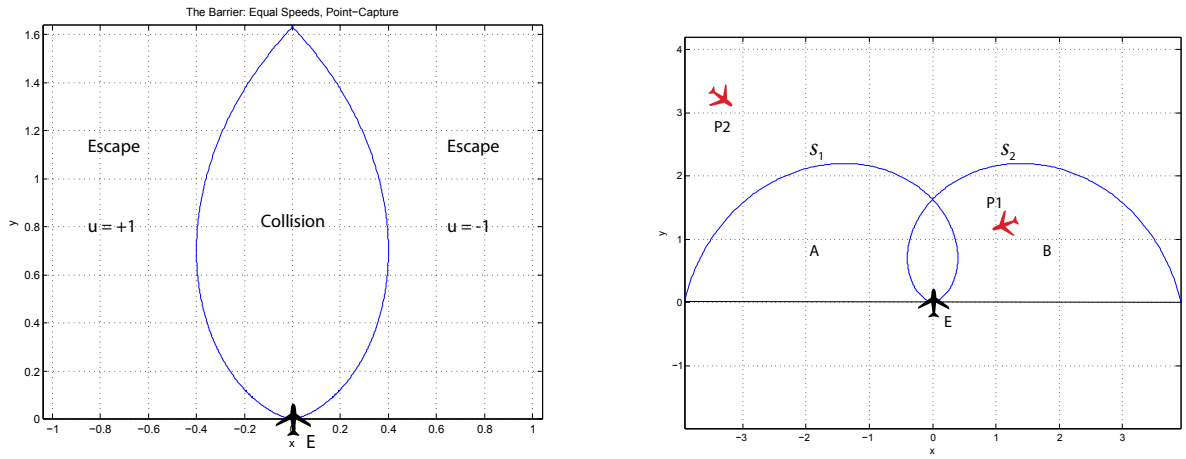


Figure 14: Evasion is guaranteed if the evader is able to construct a hyperplane separating him from the pursuers and apply his velocity vector perpendicular to that hyperplane.

states for which the evader is able to do so, as opposed to the states she is not, are separated by the barrier.

In the two-player game, the extensions of the barrier surfaces beyond $\tau_{\max}$, which corresponds to intersection on the y-axis, are of no particular significance; indeed, changing the evader strategy from $u = -1$ to $u = 1$ in the case in which the pursuer is positioned in the left quadrant instead of the right quadrant, greatly reduces the capture area (see Figure 15(a). However, in the case of multiple pursuers, and if both quadrants are populated with pursuers, these extensions do offer some important information. In Figure 15(b), consider only $\mathcal{S}_1$, the barrier if the evader chooses $u = -1$ to respond to pursuer 1. Then if pursuer 2 is outside the capture area delineated by that barrier branch, the evader has enough time to eliminate the velocity vector component pointing towards her, although she initially increases it as she turns towards pursuer 2, in order to avoid pursuer 1. By choosing $u = -1$ or $u = +1$, the evader essentially chooses the capture surface to be either the one provided by $\mathcal{S}_1$, or $\mathcal{S}_2$, respectively. We therefore retain the parts of $\mathcal{S}_1$ and $\mathcal{S}_2$ beyond $\tau_{\max}$ and thus redefine a new maximum retrograde time, say τ_m, for which equations (65) and (66) are valid, to be the first nonzero time in which the barrier curve hits the x-axis (see Figure 15(b)). Then, τ_m is the unique, non-zero solution of

$$R\sin(c\tau_m) = v\tau_m\cos(c\tau_m), \qquad \tau_m > 0. \tag{67}$$



(a) (b)

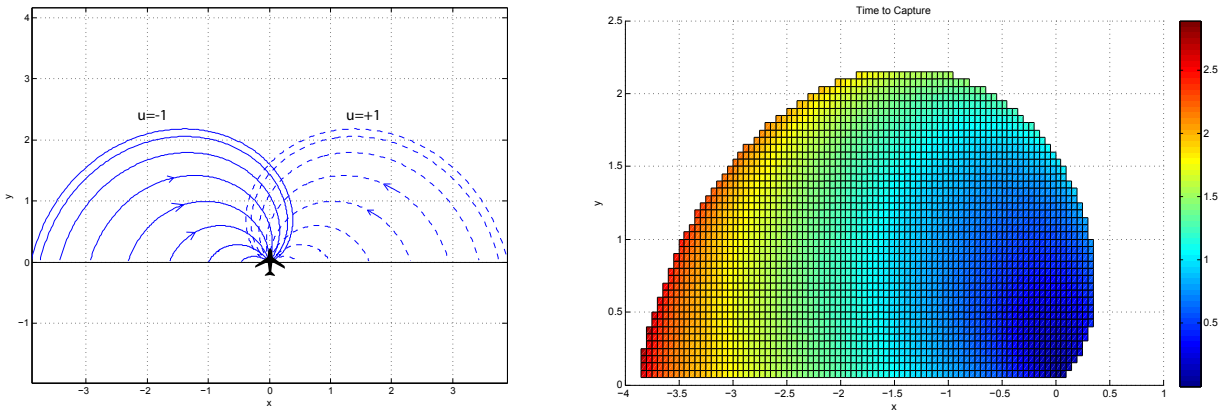
Figure 15: (a) The barrier for the case of one evader against one pursuer, given by equations (65) and (66) for $v = 1$ and $R = 0.7$. (b) The barrier surfaces if multiple pursuers are present. $\mathcal{S}_1$ separates the states that lead to capture from those that lead to evasion if the evader chooses $u = -1$, while $\mathcal{S}_2$ corresponds to $u = +1$.

The ZNP Solution and Time-to-Capture

The connection between the Suicidal Pedestrian game and the well-known Zermelo's Navigation Problem (ZNP) of optimal control has been highlighted in our previous work. Noticing that the dynamics exhibit the same form as those considered in ZNP, one may apply Zermelo's formula to calculate the rate of the optimal control of the pursuers:

$$\dot{\phi}^* = -\frac{v}{R}u, \tag{68}$$

for any fixed, constant value u of the evader control, that is, whenever u is not a function of time or state. In the two-player case, the evader's control is *bang-bang* and without switching until game termination. Thus, for $u = 1$ or $u = -1$, one may integrate the resulting system of ordinary differential equations, subject to initial conditions (x, y) and $\phi^*(0)$ that will lead to a trajectory passing through the origin $(0, 0)$. However, it is more convenient to perform this integration backwards in time, by switching the signs of the ODE's, using the origin as boundary condition and a variable retrograde boundary condition $\phi_f^* \in [0,\ 2\pi]$ for (68). This will yield a parametric family of curves, and it remains to locate the one that passes through the original point (x, y) of interest. In fact, this integration can be performed analytically to obtain the following



(a) (b)

Figure 16: (a) Several time optimal trajectories leading to capture, members of the parametric family of curves obtained by application of Zermelo's navigation law. Solid line trajectories correspond to the evader's control being $u = -1$ while dashed to $u = +1$. (b) Time to Capture (TtC), obtained by storing the τ-variable values across the optimal trajectories, and then interpolating these values over a selected grid of choice.

parametric family of curves

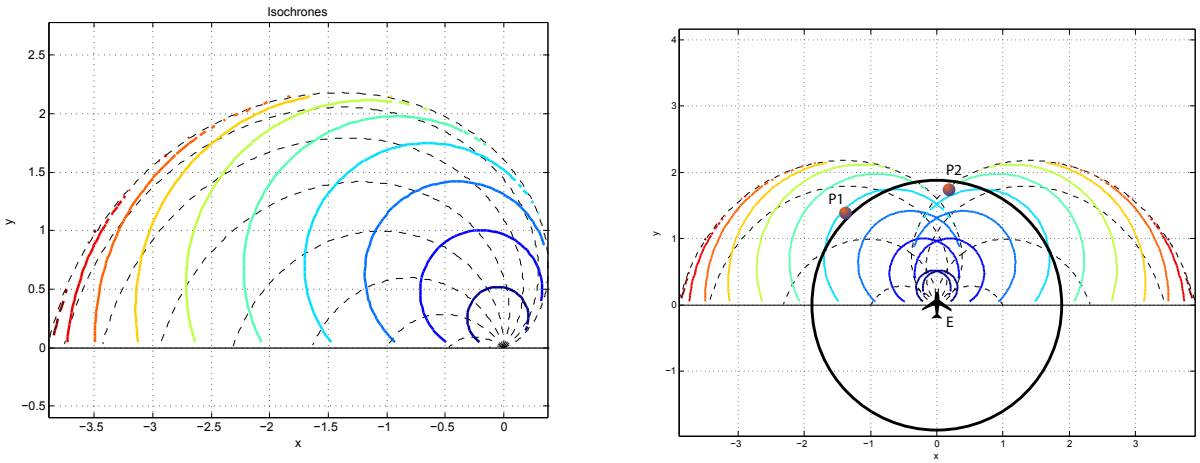
$$x(\phi_f^*; \tau) = -R + R\cos(c\tau) + v\tau\sin(\phi_f^* - c\tau), \tag{69}$$

$$y(\phi_f^*; \tau) = R\sin(c\tau) + v\tau\sin(\phi_f^* - c\tau), \qquad \tau \in [0, \tau_m], \tag{70}$$

where $u = -1$ and ϕ_f^* is the free parameter. A mere reflection on the y-axis yields the optimal trajectories if $u = +1$ is used instead. Several time optimal trajectories leading to capture are depicted in Figure 16(a), for various values of ϕ_f^*.

A further interesting fact of this connection between the SPG and the ZNP is that the parametric family of curves allows for a simple calculation of the *Time to Capture* (TtC) over the entire domain in which the evader will be intercepted, given her choice of fixed control input. To obtain a characterization of TtC as a function of the evader's fixed control input u and space, one simply has to store the τ values across the optimal trajectories, and then interpolate these values over a selected grid of choice. The result of this procedure is depicted in Figure 16(b). Note that in the two-player game, when the evader faces only one pursuer, the TtC function corresponds to the Value of the game, and can be obtained in the same manner by storing the capture time of trajectories within the two-player capture region, delineated by the barrier in Figure 15(a).

An important remark concerning collision avoidance when multiple agents are in the vicinity, and a worst-case scenario of agent hostility is assumed, is the following. By comparing the TtC functions that correspond to $u = -1$ and $u = +1$, the evader is able to prioritize between the threats posed by the surrounding adversaries. The function level sets, depicted in Figure 17(a), indicate



(a) (b)

Figure 17: (a) Isochrones –the value function level sets– when the evader's control is $u = -1$; plotted with colored lines and time optimal trajectories with dashed. (b) Although P2 is closer to the evader than P1 (indicated by a circle centered on E), P1 poses a more imminent threat as can be seen by the isochrones. Thus the evader can increase his time to capture by reacting to P1 with $u = +1$ instead of reacting to P2 with $u = -1$.

that the Euclidean distance of the pursuer relative to the evader is not the most suitable metric for the purposes of identifying which pursuer poses the most imminent threat. An example of assessing threat is shown in Figure 17(b). Although P2 is positioned closer to the evader compared to P1, as indicated by the circle centered at the evader, the evader has the incentive to react against P1 instead of P2, in order to further increase the time to capture.

Necessary and Sufficient Conditions for Evasion

Consider the multi-pursuer game described so far, in which all agents have the same constant speed and the initial positioning is such that the evader is facing all pursuers. We formally define the sets A and B as the sets of all points enclosed between $\mathcal{S}_1$ and the x-axis and $\mathcal{S}_2$ and the x-axis, respectively (see Figure 15(b)). To this end, let $\rho(\tau) \triangleq \sqrt{x^2(\tau) + y^2(\tau)}$ and substitute (65) and (66) to obtain

$$\rho(\tau) = \sqrt{2R^2(1 - \cos(c\tau)) + v^2\tau^2 - 2R\tau v \sin(c\tau)}, \qquad \tau \in [0, \tau_m], \tag{71}$$

The change of variables

$$\theta = g_1(\tau) \triangleq \arctan(y(\tau)/x(\tau)), \qquad \tau \in [0, \tau_m],\ \theta \in [0, \pi]. \tag{72}$$

yields $(\rho \circ g_1^{-1})(\theta)$ as the expression of $\mathcal{S}_1$ in polar coordinates, while the expression of $\mathcal{S}_2$ is similarly obtained by setting

$$\theta = g_2(\tau) \triangleq \arctan(-y(\tau)/x(\tau)), \qquad \tau \in [0, \tau_m],\ \theta \in [0, \pi], \tag{73}$$

to account for the reflection on the y-axis. Then, keeping in mind that the Cartesian coordinates (x, y) are transformed to polar coordinates (r, θ) through $x = r\cos\theta$, $y = r\sin\theta$, we define

$$A \triangleq \{(x, y) : x = r\cos\theta,\ y = r\sin\theta,\ \theta \in [0, \pi],\ 0 \leq r < (\rho \circ g_1^{-1})(\theta)\}, \tag{74}$$

while B is similarly defined

$$B \triangleq \{(x, y) : x = r\cos\theta,\ y = r\sin\theta,\ \theta \in [0, \pi],\ 0 \leq r < (\rho \circ g_2^{-1})(\theta)\}. \tag{75}$$

We now proceed to obtain necessary and sufficient conditions for evasion. To this end, for the sets A and B defined by (74) and (75) respectively, let

$$I_A = \begin{cases} 1, & \text{if at least one pursuer is in } A, \text{ that is, } (x_i, y_i) \in A \text{ for some i, } i = 1, \ldots, N \\ 0, & \text{otherwise.} \end{cases} \tag{76}$$

$$I_B = \begin{cases} 1, & \text{if at least one pursuer is in } B, \text{ that is, } (x_i, y_i) \in B \text{ for some i, } i = 1, \ldots, N \\ 0, & \text{otherwise.} \end{cases} \tag{77}$$

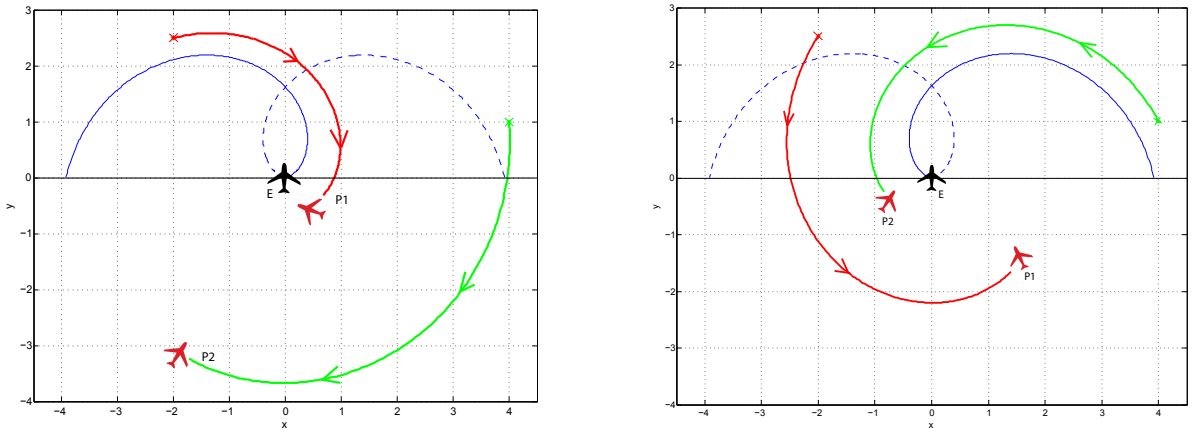
and define $\Sigma = I_A + I_B$. Notice that $\Sigma \in \{0, 1, 2\}$, and that every possible positioning of pursuing agents on the upper half of the Euclidean plane corresponds to a value in Σ.

Proposition 1. *If $\Sigma \in \{0, 1\}$, the evader escapes.*

Proof. We investigate the two cases separately:

- *Case* $\Sigma = 0$. This value corresponds to an arbitrary number of pursuers being placed outside of A and B. In this case, the evader may choose either $u = 1$ or $u = -1$ since in both cases, a time optimal trajectory leading to capture, that passes through the position of at least one of the pursuers, does not exist. This means that the evader has enough time to perform her evading maneuver. The trajectories in the evader fixed reference frame, for both cases of evader control, are depicted in Figure 18.
- *Case* $\Sigma = 1$. This implies that an arbitrary number of pursers are in *either* $A \setminus (A \cap B)$ *or* $B \setminus (A \cap B)$, but not in the intersection $A \cap B$. Similarly to the previous case, the evader has enough time to complete her turning maneuver, i.e., has a control action for which no time-optimal trajectory leading to capture passing through a pursuer location exists. The difference however is that this control action is uniquely determined ($u = 1$ for pursuers in $A \setminus (A \cap B)$ or $u = -1$ for pursuers in $B \setminus (A \cap B)$). An example of this case is depicted in Figure 15(b). The evader chooses $u = -1$ to avoid P1 and, by doing so, initially turns towards P2. However, she has enough time to complete the turn without interception by P2, and thus escapes.

□



(a) (b)

Figure 18: The case of $\Sigma = 0$. All pursuers are located outside of A and B, no ZNP solution passes through their location, and the evader can use either one of his extremal controls to escape: (a) u=-1 and (b) u=+1. In both cases, the barrier surface with a solid line corresponds to the selected control, while the one with the dashed line corresponds to the alternative choice. Both cases lead to escape. Since a policy for capture (be it optimal or not) does not exist, a pure pursuit policy is assigned to both pursuers, for the purposes of simulation.

We now proceed the next statement

Proposition 2. *If there is at least one pursuer in $A \cap B$, collision occurs.*

Proof. This follows immediately from the SPG solution. Indeed, picking an arbitrary pursuer in $A \cap B$ and disregarding all the rest of the pursuers of the game, capture is guaranteed because the pursuer is located in the capture zone delineated by the SPG, and thus the evader has no strategy that leads to evasion against that particular pursuer. Notice that this initial condition corresponds to $\Sigma = 2$. □

Before obtaining necessary and sufficient conditions for evasion, the following Lemma will be useful:

Lemma 5. *If there is at least one purser both in $A \setminus (A \cap B)$ and in $B \setminus (A \cap B)$, but none in $A \cap B$, capture occurs simultaneously by at least one pursuer from $A \setminus (A \cap B)$ and one pursuer from $B \setminus (A \cap B)$.*

Proof. It is obvious that a constant evader control $u = -1$ or $u = +1$ will lead to capture by one of the pursuers, since for both control inputs $u = -1$ and $u = +1$, a time optimal trajectory leading to capture generated by the ZNP solution and passing through pursuer positions, exists. Any intermediate constant value $u \in [-1, 1]$ will also yield capture, since the surface on the Euclidean plane covered by family of time optimal trajectories generated by the ZNP formula, is minimal for the extreme evader control values; any intermediate constant value of u will yield sets A' and B' which are supersets of A and B respectively. The singular case $u = 0$ is no exception, since Zermelo's law for the pursuers then reduces to *parallel navigation*, which again leads to interception.

Simultaneous capture occurs because of the symmetry of the problem at hand. To this end, assume, *ad absurdum* that the evader gets intercepted by only one pursuer, namely the one on his right (without loss of generality). Then, just before interception occurs, the evader could prolong his time to capture by turning to the left, towards the other pursuer. Thus, an optimal evader maneuver leads to simultaneous capture by at least one pursuer of each domain. □

Noticing that the initial condition described in Lemma 1 corresponds to $\Sigma = 2$ and combining Proposition 1, Proposition 2 and Lemma 1, the following corollary is obtained immediately.

Corollary 2. *Collision occurs if and only if* $\Sigma = 2$.

Corollary 1 summarizes the necessary and sufficient conditions for evasion, under Assumption 1, in terms of purser initial positioning on the plane. It demonstrates that, once the initial positions of all pursuers have been specified, the game outcome is determined. Furthermore, Corollary 1 defines which areas of the Euclidean plane are considered "danger zones", in the sense that placing at least one pursuer at that location leads to a different game outcome.

3.2 Optimal Evader Strategies in a Two-Pursuer One-Evader Problem

The problem under investigation in this section was motivated by the following situation: consider a decoy, whose speed is limited, entering a defense area guarded by multiple agents following a prescribed pursuing protocol. The decoy's objective is to avoid capture as long as possible so that it can "buy more time" for the other evader(s). The goal is to find the optimal evading strategy to maximize the time-of-capture.

We assumed a simplified version of this problem, involving only two-pursuers and one evader. Both pursuers implement a *relay-pursuit strategy*, according to which only one pursuer is assigned to go after the target at every instant of time. The latter is the *active* pursuer, while the other pursuer is designated as the *inactive* pursuer. This strategy may be desirable in cases when the pursuing agents play a dual role, namely, both as pursuers and as guardians protecting some area of interest, or when the power or energy consumption of the pursuing agents needs to be taken into consideration.

In our problem formulation, we also assumes that the pursuers have a stroboscopic view of the evader's position, i.e., at every instant of time, the pursuers only know the current position of the evader but not its velocity. It is further assumed that the evader is slower than the two pursuers, but is aware of their strategies and current positions.

Problem Formulation

Due to the symmetry of the problem, when the evader resides on the Voronoi cell boundary, we can assign any one of the two pursuers to be the active pursuer. Therefore, throughout the pursuit process, we can fix one pursuer to be the active pursuer, while the inactive pursuer remains stationary on the plane, whose mere presence, however, imposes a state restriction, namely that the evader does not enter the interior of its corresponding Voronoi cell.

The equivalent one-pursuer one-evader optimal control problem with state constraint can be expressed in a reduced three-dimensional space by fixing the origin of a new coordinate system to the active pursuer and by aligning the positive direction of the x-axis with the direction of the active pursuer's velocity vector. Let us denote this frame by $\mathcal{M}$. In the frame $\mathcal{M}$, the active pursuer is fixed at the origin, and the motion of the evader is restricted to lie on the x-axis due to the pure pursuit strategy of the active pursuer. The inactive pursuer, however, is no longer stationary in $\mathcal{M}$.

Let $\theta = \phi_E - \phi_P$ be the angle between the active pursuer's and evader's velocity vectors. Then $\dot{\phi}_P$ is given by

$$\dot{\phi}_P = \frac{v \sin\theta}{\|\mathsf{x}_E - \mathsf{x}_P\|}. \tag{78}$$

Let the coordinates of the inactive pursuer and the evader be, respectively, $\mathsf{x}_G = (\eta, \zeta)$ and $\mathsf{x}_E = (\chi, 0)$ in the frame $\mathcal{M}$, as shown in Figure 19. The equations of motion in the frame $\mathcal{M}$ are then given by

$$\dot{\chi} = -u + v\cos\theta, \tag{79}$$

$$\dot{\eta} = -u + v\frac{\zeta}{\chi}\sin\theta, \tag{80}$$

$$\dot{\zeta} = -v\frac{\eta}{\chi}\sin\theta, \tag{81}$$

while the constraint is given by

$$\mathcal{S}(\xi) = \frac{1}{2}(\chi^2 - (\eta - \chi)^2 - \zeta^2) \leq 0, \tag{82}$$

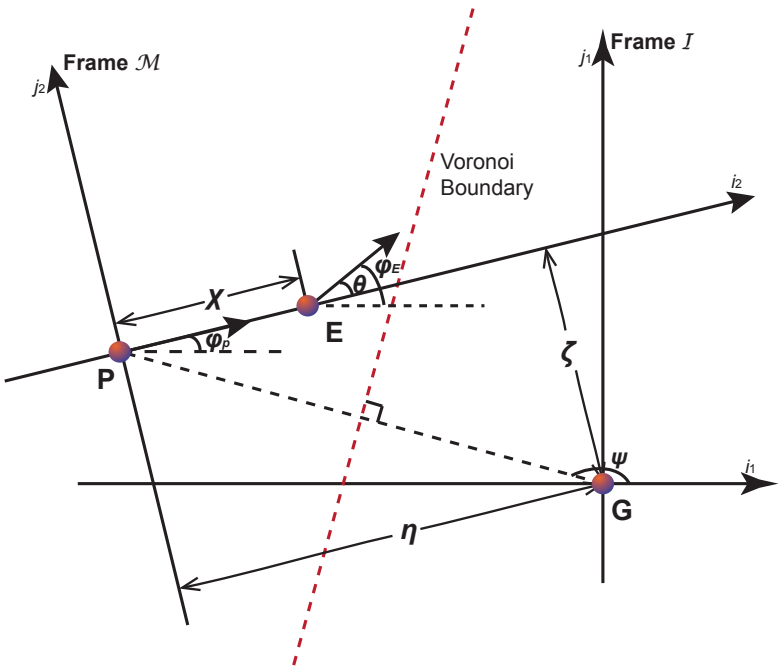


Figure 19: Geometry of evader (E), active (P) pursuer, and inactive (G) pursuer in the inertial ($\mathcal{I}$) and the active pursuer frames ($\mathcal{M}$).

where $\xi = [\chi, \eta, \zeta]^\mathsf{T} \in \mathbb{R}^3$. The boundary conditions in frame $\mathcal{M}$ are given by

$$\begin{aligned} &\chi(0) = \|\mathsf{x}_E(0) - \mathsf{x}_P(0)\|, \quad \chi(t_f) = \epsilon, \\ &\eta(0) = \|\mathsf{x}_P(0)\| \cos(\pi + \psi(0) - \phi_P(0)), \quad \eta(t_f) \text{ free}, \\ &\zeta(0) = \|\mathsf{x}_P(0)\| \sin(\pi + \psi(0) - \phi_P(0)), \quad \zeta(t_f) \text{ free}, \end{aligned} \tag{83}$$

where $\psi(0)$ is the initial value of the angle $\psi = \text{atan}(y_P, x_P)$, as shown in Figure 19. Given (79)-(81), the constraint (82) and boundary conditions (83), our goal is to find the optimal control θ to maximize t_c.

Problem Analysis and Main Results

Consider the optimal control problem (79)-(83). Assume that $u > v > u/2$ and assume that the initial conditions are such that the constraint is active. Utilizing the Pontryagin's Minimum

Principle, our analysis has shown that the optimal control of the evader is given as follows:

$$\tan\theta^*(t) = \begin{cases} \dfrac{\lambda_2\zeta - \lambda_3\eta}{\lambda_1\chi}, & t \in [0, \tau_1], \\ \dfrac{q - \sigma p\sqrt{p^2+q^2-1}}{p + \sigma q\sqrt{p^2+q^2-1}}, & t \in [\tau_1, \tau_2], \\ 0, & t \in [\tau_2, t_c], \end{cases}$$

where $p = v\eta/(u\chi), q = v\zeta/(u\chi), \sigma = \mathrm{sgn}(q)$. Furthermore, τ_2 satisfies the switching condition:

$$v\eta(\tau_2) - u\chi(\tau_2) = 0. \tag{84}$$

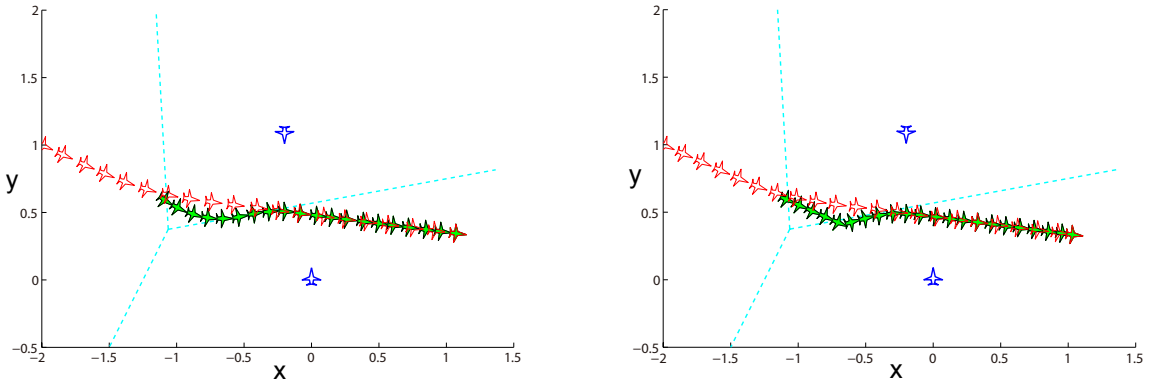
Summarizing the previous analysis, we conclude that the optimal trajectory of the evader involves three periods: first she moves in the Voronoi cell of the active pursuer in a way such that the optimal conditions (transversality condition, Erdmann corner condition, etc.) are satisfied before she hits the boundary, then she moves along the boundary until the switching condition (84) is satisfied, finally the evader moves along the LoS till capture occurs.

Given the analysis of the previous section, one can compute numerically the evader's optimal trajectory. Note, however, that the second and third period strategies can not be easily implemented without resorting to the solution of a two-point boundary value problem. Therefore, we hereby propose a suboptimal control scheme for the evader that can be computed analytically. It is presented as follows: The evader first moves along the LoS away from the pursuer before it hits the boundary. Then she moves on the boundary until the switching condition (84) is satisfied. Afterwards, the evader resumes moving along the LoS until capture occurs. Our simulation results show that the relative differences between the time-to-capture by applying this strategy and the optimal one are always less than 5%, given different speed ratio and initial positions of the evader and the pursuer.

The main idea of the proposed suboptimal evader control strategy is that the evader moves along the LoS whenever it is able to, otherwise it will move along the Voronoi cell boundary. We can further generalize this idea and propose a suboptimal evading strategy for the multiple-pursuer/one-evader relay pursuit problem. In this case, there will be one active pursuer and multiple inactive pursuers that play the role of guards. The objective of the evader is to maximize the instantaneous velocity component along the LoS at every instant of time without violating the state constraints imposed by the fact that the evader never enters the interior of the Voronoi cells of an inactive pursuer.

We have implemented the generalization of the suboptimal evading strategy to a three-pursuer/one-evader relay pursuit problem and compared the result with the numerical optimal control of the evader. The numerical result is shown in Figure 20. The corresponding optimal time-to-capture is $t_c = 3.1530$. The evader's trajectory generated by applying the evading strategy we proposed, and the active pursuer's trajectory are presented in Figure 20 in red and green lines, respectively.

As seen from Figures 20(a) and 20(b) the trajectories are very similar. The time-to-capture is $t_c' = 3.1366$. The relative time difference this time is $\Delta = 0.52\%$.



(a) (b)

Figure 20: (a) Trajectory of the active pursuer (in red) and optimal trajectory of the evader (in green) obtained from GPOPS. The cyan line represents the Voronoi diagram of the three pursuers with initial conditions. (b) Trajectories generated by applying the proposed suboptimal evading strategy.

3.3 Pursuit Evasion Game of Two Players under an External Flow Field

One common assumption among all previous results on pursuit and evasion has been that no environmental conditions may affect the outcome of the game. However, when either the pursuer or the evader (or both) is a small autonomous underwater vehicle (AUV) or an unmanned aerial vehicle (UAV), the presence of the sea current or the wind, will significantly affect their motion. As a result, their behavior and the solution of the differential game may be greatly affected by the existence of the external flow field. To fill this gap in the literature, we considered the differential game of pursuit and evasion between two players on a plane under an external flow field. It is assumed that the pursuer and the evader move with constant but different speeds, and they are both agile, that is, they are allowed to change their headings instantaneously. To simplify the analysis, it was assumed that the flow field is approximated by a time-invariant, spatially-affine function. Our main goal was to find the region of initial conditions of both players that leads to capture when both players act optimally, and derive the corresponding optimal strategies of the two players when capture is guaranteed.

Problem Formulation

Consider a pursuer and an evader moving on a plane under the influence of an external flow field. The equations of motion for the pursuer and the evader in the inertial reference frame are given by

$$\dot{x}_P = v_P \cos\phi + w_1(x_P, y_P), \tag{85}$$
$$\dot{y}_P = v_P \sin\phi + w_2(x_P, y_P), \tag{86}$$
$$\dot{x}_E = v_E \cos\psi + w_1(x_E, y_E), \tag{87}$$
$$\dot{y}_E = v_E \sin\psi + w_2(x_E, y_E), \tag{88}$$

where $(x_P, y_P) \in \mathbb{R}^2$ and $(x_E, y_E) \in \mathbb{R}^2$ denote the positions of the pursuer and the evader, respectively, $\phi, \psi \in [-\pi, \pi]$ are the control of the pursuer and the evader, and v_P and v_E represent the speed of the pursuer and the evader, respectively. In this work, we will assume that $v_P > v_E$. Finally, $w_1(\cdot,\cdot)$ and $w_2(\cdot,\cdot)$ are the components of an external spatially varying flow field along x-axis and y-axis, respectively.

In order to simplify the analysis, it will be assumed that the external flow field is approximated, at least locally, by a time-invariant affine function. Specifically, let

$$w_1(x, y) = \alpha_1 x + \beta_1 y + \gamma_1, \tag{89}$$
$$w_2(x, y) = \alpha_2 x + \beta_2 y + \gamma_2, \tag{90}$$

where α_i, β_i, $\gamma_i \in \mathbb{R}$ $(i = 1, 2)$ are prescribed constants. By choosing a new reference frame whose origin is at the pursuer, the kinematic equations can be represented in a reduced two-dimensional space space. In particular, let $x = x_E - x_P$ and $y = y_E - y_P$ be the relative distance between the evader and the pursuer along the x-axis and y-axis, respectively. The kinematic equations in terms of x and y are then given by

$$\dot{x} = v_E \cos\psi - v_P \cos\phi + \alpha_1 x + \beta_1 y, \tag{91}$$
$$\dot{y} = v_E \sin\psi - v_P \sin\phi + \alpha_2 x + \beta_2 y. \tag{92}$$

By defining the reduced state as $\mathsf{x} = [x, y]^\mathsf{T}$, the equations can then be written compactly as

$$\dot{\mathsf{x}} = v_E \mathbf{v} - v_P \mathbf{u} + w(\mathsf{x}), \tag{93}$$

where $\mathbf{v} = [\cos\psi, \sin\psi]^\mathsf{T}$ and $\mathbf{u} = [\cos\phi, \sin\phi]^\mathsf{T}$ are the controls, and where the relative wind field is given by $w(\mathsf{x}) = A\mathsf{x}$, where $A = \begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{bmatrix}$. The game terminates when capture occurs, that is, when the evader is in the *interior* of a ball $\mathcal{B}$ of radius ℓ centered at the pursuer's current location, given by $\mathcal{B} = \{\mathsf{x} \in \mathbb{R}^2 : |\mathsf{x}| \leq \ell\}$. The *terminal surface* is the manifold in the state space which, once penetrated, determines termination of the game. The terminal surface $\mathcal{C}$ is thus the circle centered at the origin of radius ℓ, i.e., $\mathcal{C} = \{\mathsf{x} \in \mathbb{R}^2 : |\mathsf{x}| = \ell\}$. Accordingly, the state space $\mathcal{E}$ is the portion of the x, y-plane exterior to $\mathcal{C}$, that is, $\mathcal{E} = \{\mathsf{x} \in \mathbb{R}^2 : |\mathsf{x}| > \ell\}$.

We want to find the region in the state space such that the evader can be captured by the pursuer if their initial relative coordinates fall inside this region. This region is denoted as the *capture zone*. The region which leads to escape of the evader is the *escape zone*.

Problem Analysis

To analyze the problem we followed the standard approach of the game of kind of Isaacs. First, we focused on identifying the *usable part* of the terminal surface. The UP is the subset of $\mathcal{C}$ in which the pursuer can cause termination immediately when both players act optimally. The remaining points on $\mathcal{C}$ form the nonuseable part, that is, termination will not occur even if the trajectory reaches this part of $\mathcal{C}$ under optimal play (i.e., when both players act optimally). The part of $\mathcal{C}$ that separates the usable part and the nonuseable part of $\mathcal{C}$ is called the *boundary of the usable part* (BUP).

In order to find the usable part, we parameterize $\mathcal{C}$ with the variable s according to

$$x = \ell \cos s, \qquad y = \ell \sin s. \tag{94}$$

Let $r^2 = x^2 + y^2 = |\mathsf{x}|^2$. Taking the time derivative on both sides of the last equation, and consider only points on $\mathcal{C}$, we have

$$\begin{aligned} \ell \dot{r} = {} & \ell \cos s (v_E \cos\psi - v_P \cos\phi + \alpha_1 \ell \cos s + \beta_1 \ell \sin s) \\ & + \ell \sin s (v_E \sin\psi - v_P \sin\phi + \alpha_2 \ell \cos s + \beta_2 \ell \sin s). \end{aligned}$$

The usable part of $\mathcal{C}$ is specified by the condition

$$\min_{\phi} \max_{\psi} \dot{r}(\mathsf{x}) < 0, \quad \mathsf{x} \in \mathcal{C}, \tag{95}$$

which implies that the relative trajectory is able to penetrate the terminal surface $\mathcal{C}$. From the last two inequalities, and using standard trigonometric identities, we have that, for $\mathsf{x} \in \mathcal{C}$,

$$\begin{aligned} \min_{\phi} \max_{\psi} \ \dot{r}(\mathsf{x}) = {} & v_E - v_P + \frac{\ell}{2}(\alpha_1 + \beta_2) \\ & + \frac{\ell}{2}[(\alpha_1 - \beta_2)\cos 2s \quad + (\beta_1 + \alpha_2)\sin 2s]. \end{aligned} \tag{96}$$

Let now $\sigma = \sqrt{(\alpha_1 - \beta_2)^2 + (\beta_1 + \alpha_2)^2}$. When $\sigma = 0$, we have $\alpha_1 - \beta_2 = 0$ and $\beta_1 + \alpha_2 = 0$. Hence, whether the game terminates depends on the sign of $v_E - v_P + \ell(\alpha_1 + \beta_2)/2$. Specifically, when $v_E - v_P + \ell(\alpha_1 + \beta_2)/2 < 0$, the usable part of the terminal surface is $\mathcal{C}$ itself, whereas when $v_E - v_P + \ell(\alpha_1 + \beta_2)/2 > 0$, the game will not terminate under any initial conditions of the pursuer and the evader, which means that the evader always escapes. In the latter case the whole state space $\mathcal{E}$ is the escape zone.

Henceforth, we assume that $\sigma \neq 0$. Then (95) and (96) imply that

$$\min_{\phi} \max_{\psi} \dot{r} = v_E - v_P + \frac{\ell}{2}(\alpha_1 + \beta_2) + \frac{\ell\sigma}{2} \sin(\theta + 2s) < 0, \tag{97}$$

where θ is given by $\sin\theta = (\alpha_1 - \beta_2)/\sigma$ and $\cos\theta = (\beta_1 + \alpha_2)/\sigma$. From (97) we reach the conclusion that in the reduced space the game will *not* terminate if

$$\frac{2(v_P - v_E) - \ell(\alpha_1 + \beta_2)}{\ell\sigma} < -1, \tag{98}$$

where $\sigma = \sqrt{(\alpha_1 - \beta_2)^2 + (\beta_1 + \alpha_2)^2}$. Finally, when $\zeta \geq 1$ the usable part is the whole terminal surface $\mathcal{C}$.

Henceforth, we assume that $-1 \leq \zeta < 1$. Under this assumption, the BUP is determined from $\sin(\theta + 2s) = \zeta$. This yields four solutions in $[0, 2\pi)$, denoted by s_1, s_2, $s_3 = s_1 + \pi$, $s_4 = s_2 + \pi$. Hence, the BUP contains four points on $\mathcal{C}$, represented by $P_i = (\cos s_i, \sin s_i)$, $i = 1, \ldots, 4$. A typical illustration of the terminal surface, which is divided into the usable and nonusable parts by the BUP, which consists of four points on the terminal surface, is shown in Figure 21.

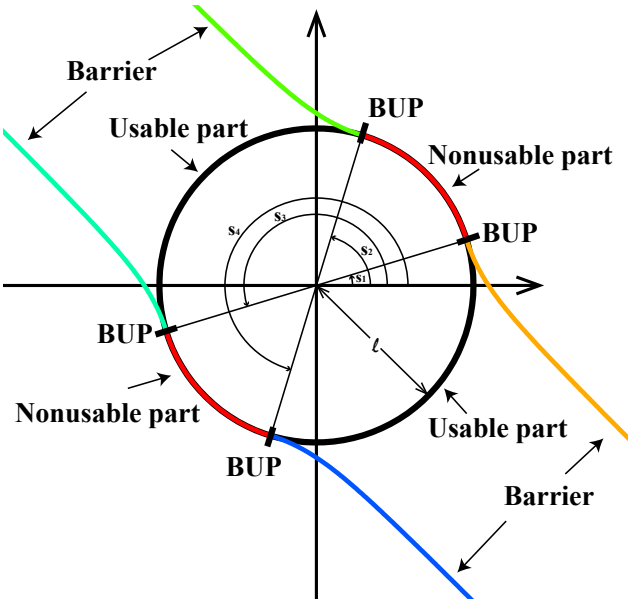


Figure 21: The terminal surface $\mathcal{C}$ of the game is given by a circle of radius ℓ. The circle is separated by the BUP (4 points on the circle parameterized by s_1 through s_4) into the usable part (black lines) and the nonusable part (red lines). Every barrier meets the terminal surface at the BUP tangentially.

Now we turn to the construction of the *barrier*. The barrier is a surface in the state space that consists of initial conditions for which the outcome is neutral. One property of the barrier is that it

is never crossed by either the pursuer or the evader during optimal play. In particular, the barrier emanates from the BUP and is tangent to $\mathcal{C}$ at the BUP. We denote the barrier by $\mathcal{S}$. At each point on $\mathcal{S}$ we define the normal vector $\nu = [\nu_1, \nu_2]^\mathsf{T} \in \mathbb{R}^2$ extending into the escape zone.

Denoting with $(\dot{})$ the derivative with respect to τ, the retrograde evolution of the states and the vector ν can be established as:

$$\begin{aligned}
\mathring{x} &= (v_P - v_E)\frac{\nu_1}{\rho} - \alpha_1 x - \beta_1 y, \\
\mathring{y} &= (v_P - v_E)\frac{\nu_2}{\rho} - \alpha_2 x - \beta_2 y, \\
\mathring{\nu}_1 &= \alpha_1 \nu_1 + \alpha_2 \nu_2, \\
\mathring{\nu}_2 &= \beta_1 \nu_1 + \beta_2 \nu_2.
\end{aligned} \tag{99}$$

By the definition of the BUP and the barrier, it is clear that the barrier starts at the BUP towards the state space $\mathcal{E}$ in a retrogressive sense. Moreover, the two surfaces meet tangentially, since no penetration occurs at the BUP and the vectorfields of both players are tangential to the barrier. Pick any $\bar{s} \in \{s_1, s_2, s_3, s_4\}$. The initial conditions for the RPEs are thus given by $x(\tau = 0) = \ell \cos \bar{s},\ y(\tau = 0) = \ell \sin \bar{s},\ \nu_1(\tau = 0) = \cos \bar{s},\ \nu_2(\tau = 0) = \sin \bar{s}$. By integrating (99) subject to these initial conditions we obtain

$$\nu(\tau) = e^{A^\top \tau} \nu(\tau = 0), \tag{100}$$

$$\mathsf{x}(\tau) = e^{-A\tau} \mathsf{x}(\tau = 0) + \int_0^\tau e^{-A(\tau - \xi)} b(\xi)\, \mathrm{d}\xi, \tag{101}$$

where $b(\tau) = (v_P - v_E)\nu(\tau)/|\nu(\tau)|$. Therefore,

$$\mathsf{x}(\tau) = \ell e^{-A\tau} \begin{bmatrix} \cos \bar{s} \\ \sin \bar{s} \end{bmatrix} + (v_P - v_E) e^{-A\tau} \int_0^\tau \frac{e^{(A + A^\top)\xi}}{|\nu(\xi)|} \begin{bmatrix} \cos \bar{s} \\ \sin \bar{s} \end{bmatrix} \mathrm{d}\xi. \tag{102}$$

By plotting the trajectories of (102) given the four initial conditions of the BUP, we can determine whether these are valid barriers and whether the state space $\mathcal{E}$ is separated by the barriers.

Game of Degree in the Capture Region

Now that we have identified the capture region, we aim at determining the optimal trajectory of x inside this region by solving a game of degree. Within the capture region, the performance index is $\mathcal{J} = \int_0^{t_c} \mathrm{d}t$. To this end, we define the value function $\mathbf{V}(\mathsf{x})$, which satisfies the Hamilton-Jacobi-Isaacs (HJI) equation

$$0 = \min_{\phi} \max_{\psi} \mathbf{H}(\mathsf{x}, \mathbf{V}_\mathsf{x}), \tag{103}$$

where the Hamiltonian $\mathbf{H}$ is given by

$$\begin{aligned}\mathbf{H} = 1 &+ \frac{\partial \mathbf{V}}{\partial x}\Big(v_E \cos\psi - v_P \cos\phi + \alpha_1 x + \beta_1 y\Big) \\ &+ \frac{\partial \mathbf{V}}{\partial y}\Big(v_E \sin\psi - v_P \sin\phi + \alpha_2 x + \beta_2 y\Big).\end{aligned} \tag{104}$$

Let $\mathbf{V}_x = \frac{\partial \mathbf{V}}{\partial x}$, $\mathbf{V}_y = \frac{\partial \mathbf{V}}{\partial y}$. Then (103) can be rewritten as

$$\begin{aligned}0 =& 1 + \mathbf{V}_x(\alpha_1 x + \beta_1 y) + \mathbf{V}_y(\alpha_2 x + \beta_2 y) \\ &+ \min_{\phi}\{-v_P(\mathbf{V}_x \cos\phi + \mathbf{V}_y \sin\phi)\} \\ &+ \max_{\psi}\{v_E(\mathbf{V}_x \cos\psi + \mathbf{V}_y \sin\psi)\}.\end{aligned} \tag{105}$$

Hence, the optimal controls ϕ^* and ψ^* are given by

$$\begin{aligned}\cos\phi^* &= \frac{\mathbf{V}_x}{\mu}, \qquad \sin\phi^* = \frac{\mathbf{V}_y}{\mu}, \\ \cos\psi^* &= \frac{\mathbf{V}_x}{\mu}, \qquad \sin\psi^* = \frac{\mathbf{V}_y}{\mu},\end{aligned} \tag{106}$$

where $\mu = \sqrt{\mathbf{V}_x^2 + \mathbf{V}_y^2}$. Plugging (106) back into the Hamiltonian, we get the optimal Hamiltonian

$$\mathbf{H}^* = 1 + (v_E - v_P)\mu + \mathbf{V}_x(\alpha_1 x + \beta_1 y) + \mathbf{V}_y(\alpha_2 x + \beta_2 y). \tag{107}$$

The RPEs can then be expressed as

$$\mathring{x} = (v_P - v_E)\frac{\mathbf{V}_x}{\mu} - \alpha_1 x - \beta_1 y, \tag{108}$$

$$\mathring{y} = (v_P - v_E)\frac{\mathbf{V}_y}{\mu} - \alpha_2 x - \beta_2 y, \tag{109}$$

$$\mathring{\mathbf{V}}_x = \alpha_1 \mathbf{V}_x + \alpha_2 \mathbf{V}_y, \tag{110}$$

$$\mathring{\mathbf{V}}_y = \beta_1 \mathbf{V}_x + \beta_2 \mathbf{V}_y. \tag{111}$$

On the terminal surface $\mathcal{C}$, we have $\mathbf{V} = 0$. Along with the parameterization of $\mathcal{C}$ by $x = \ell\cos s$, $y = \ell \sin s$, we get

$$0 = \frac{\partial \mathbf{V}}{\partial s} = \ell(-\mathbf{V}_x \sin s + \mathbf{V}_y \cos s).$$

Upon solving these equations, we further get, for some $\delta > 0$,

$$\mathbf{V}_x(\tau = 0) = \delta \cos s, \qquad \mathbf{V}_y(\tau = 0) = \delta \sin s. \tag{112}$$

By substituting (112) into the expression for $\mathbf{H}^*$, one can solve for δ to obtain $\delta = 1/\eta$ where $\eta = v_P - v_E - \ell(\alpha_1 \cos^2 s + (\beta_1 + \alpha_2) \sin s \cos s + \beta_2 \sin^2 s)$. Integrating the RPE's (108) through (111) subject to the initial conditions (112) yields

$$\begin{bmatrix} \mathbf{V}_x(\tau) \\ \mathbf{V}_y(\tau) \end{bmatrix} = e^{A^\top \tau} \begin{bmatrix} \delta \cos s \\ \delta \sin s \end{bmatrix}, \tag{113}$$

and hence

$$\begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix} = e^{-A\tau} \begin{bmatrix} \ell \cos s \\ \ell \sin s \end{bmatrix} + (v_P - v_E) e^{-A\tau} \int_0^\tau \frac{e^{(A+A^\top)\xi}}{\mu(\xi)} \begin{bmatrix} \delta \cos s \\ \delta \sin s \end{bmatrix} \mathrm{d}\xi, \tag{114}$$

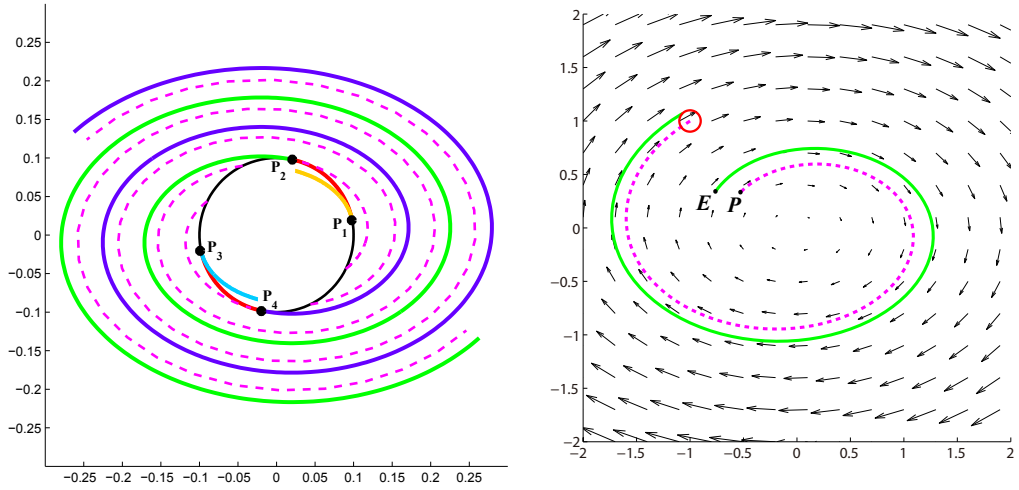
which yields the optimal trajectory for the game of degree.

Simulation Results

In this section, we present numerical simulations to illustrate the previous analysis. In the following cases, we vary the matrix A while we keep ℓ, v_P and v_E fixed to compute different types of barriers under different flow fields. Henceforth, we let $\ell = 0.1$, $v_P = 1.0$, $v_E = 0.9$. For the parameters of the flow field in the inertial frame, we set $\gamma_1 = \gamma_2 = 0$.

Case 1: $A = \begin{bmatrix} 0 & 10 \\ -5 & 0 \end{bmatrix}$. This matrix has two pure imaginary eigenvalues (center). In this case, $\sigma = 5$, $\zeta = 0.4$ and the corresponding values for the BUP are $s_1 = 0.2058, s_2 = 1.3650, s_3 = 3.3474$ and $s_4 = 4.5066$. As shown in Figure 22(a), the trajectories of the RPEs emanating from P_1 and P_3 are inside $\mathcal{B}$; these two trajectories are outside the state space $\mathcal{E}$. Hence, they are not valid barriers and are discarded. On the other hand, the trajectories emanating from P_2 and P_4 are valid barriers. They have spiral-like shapes but they fail to separate the state space into two parts. The whole state space is a capture zone; regardless of the initial conditions of the two players, capture is guaranteed. The dashed magenta lines in Figure 22(a) show the optimal trajectories in relative coordinates with respect to different initial positions on the usable part of the terminal surface. Although the barrier does not separate the state space into capture and escape zones, it is still not crossed during optimal play, which gives us some information as to how the optimal trajectories look like. The barrier also marks a discontinuity in the value function.

Given the initial positions for the evader and the pursuer as $\mathsf{x}_E(0) = [-0.764, 0.337]$ and $\mathsf{x}_P(0) = [-0.524, 0.336]$, respectively, the optimal trajectories of the evader and the pursuer in the inertial frame are depicted in Figure 22(b). These trajectories are consistent with the external flow field represented by the black arrows. The results suggest that both players are trying to take advantage of the flow field. Intuitively, this makes sense. Since the matrix A has purely imaginary eigenvalues, the uncontrolled system trajectories are circles around the origin. The flow field does not give an advantage to either the pursuer or the evader. It is then reasonable that under optimal controls



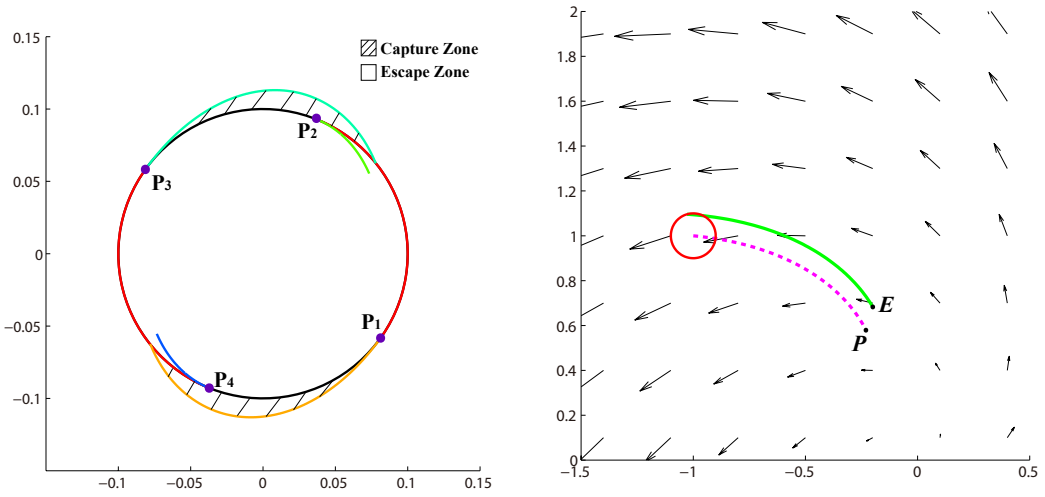
(a) (b)

Figure 22: (a) Barriers in frame $\mathcal{M}$ when $A = [0, 10; -5, 0]$. The dashed magenta lines are the optimal trajectories of the relative coordinates emanating from the usable part of the terminal surface. (b) Optimal trajectories of the evader in green and the pursuer in dashed magenta, respectively, where $A = [0, 10; -5, 0]$. Red circle around the final position of the pursuer represents the terminal surface. The flow field is depicted by the black arrows in the background.

of both players, the trajectories in the reduced state move in spiral-like patterns, as confirmed in Figure 22(a).

Case 2: $A = \begin{bmatrix} 1.4020 & -1.0772 \\ 1.4770 & 0.7756 \end{bmatrix}$. In this case, the eigenvalues are a complex conjugate pair with positive real part (unstable spiral). These values correspond to $\sigma = 0.7431$, $\zeta = -0.2390$ and the corresponding parameters for the BUP are $s_1 = 5.6612, s_2 = 1.1901, s_3 = 2.5196, s_4 = 4.3317$. As depicted in Figure 23(a), and similarly to the first case, the trajectories of the RPEs emanating from P_2 and P_4 are inside $\mathcal{B}$ and are thus discarded. The trajectories starting from P_1 and P_3 intersect $\mathcal{C}$ after some time, and thus the trajectories after the intersection are discarded. In this case, the barrier separates the capture zone from the escape zone. The capture zone is represented by the shaded region in Figure 23(a). All the remaining space outside the circle is the escape zone. The optimal trajectories of the evader and the pursuer in the inertial frame are depicted in Figure 23(b) with initial positions $\mathsf{x}_E(0) = [-0.20, 0.683]$ and $\mathsf{x}_P(0) = [-0.230, 0.579]$, respectively. Notice that in this case, there is a small region of relative initial positions for the pursuer and the evader such that capture occurs.

In this case the matrix A has two complex eigenvalues with positive real parts, which implies that the origin is an unstable spiral. The trajectories of the uncontrolled system would result in



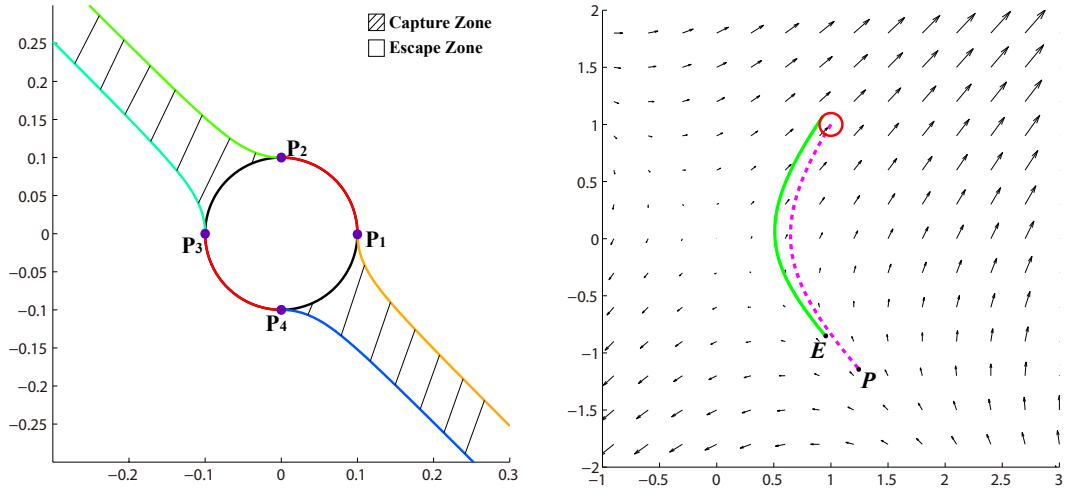
(a) (b)

Figure 23: (a) Barriers in frame $\mathcal{M}$ for $A = [1.4020, -1.0772; 1.4770, 0.7756]$, the shaded region is the escape zone and the white region outside the circle is the escape zone. (b) Optimal trajectories of the evader in green and the pursuer in dashed magenta, for $A = [1.4020, -1.0772; 1.4770, 0.7756]$.

$|x| \to \infty$ as time goes on. In this case, the flow field gives an advantage to the evader. Indeed, as shown in Figure 23(a), the capture zone is very small compared to the escape zone.

Case 3: $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$. In this case, $\sigma = 4$, $\zeta = 0$ and the corresponding parameters for the BUP are $s_1 = 0, s_2 = \pi/2, s_3 = \pi$ and $s_4 = 3\pi/2$. As illustrated in Figure 24(a), all four trajectories emanating from P_1, P_2, P_3 and P_4 are valid barriers. They separate the state space into two capture zones and two escape zones, depicted in the figure by the two shaded regions and the two white regions, respectively. Typical optimal trajectories of the evader and the pursuer in the inertial frame with initial positions $\mathsf{x}_E(0) = [0.951, -0.852]$ and $\mathsf{x}_P(0) = [1.265, -1.165]$ are shown in Figure 24(b).

In Cases 3, the matrix A has one positive eigenvalue and one negative eigenvalue. Hence, the origin is a saddle point, and in some part of the plane the flow field points towards the origin (helping the pursuer), whereas in other parts it points away from the origin (thus giving an advantage to the evader), as indicated by the black vector fields in Figure 24(b). This suggests that the pursuer tries to steer the game in the part of the space that the flow field is beneficial to him and the evader does the same, i.e., tries to steer the state to the parts of the state space that are more helpful to him. In this case, the game will terminate (or not) depending on whether the pursuer can capture the evader before the latter moves in the part of the space that the former has an advantage.



(a) (b)

Figure 24: (a) Four valid barriers in frame $\mathcal{M}$ emanating from P_1 through P_4, for $A = [1, 2; 2, 1]$. The state space is divided by two shaded capture zones and two white escape zones. (b) Optimal trajectories of the evader in green and the pursuer in dashed magenta, for $A = [1, 2; 2, 1]$.

3.4 Sequential Pursuer-Target Assignment Problem Under External Disturbances

Consider a scenario where a group of helicopters or small UAVs in a wind field are trying to capture a vehicle moving on the ground, or a team of small marine or underwater vehicles attempting to reach a ship which is large enough so that the sea currents do not significantly affect its motion. Given such a group of pursuers, we want to find a pursuit strategy to intercept the target in minimum time. Problems of this nature fall under the general class of group pursuit problems. These are difficult problems to solve, in general. Their solution is also based on the information the pursuers and the target have about each other, resulting in either cooperative or non-cooperative strategies. In this work, in order to solve this problem, we propose a *sequential* pursuit strategy. By sequential (or relay) pursuit we mean that only one pursuer is assigned to chase the target at every instant of time. In addition to simplifying significantly the group pursuit problem, a relay pursuit strategy may be desirable in cases where the power consumption of the agents is an important factor, when the agents also play a dual role as guardians protecting a certain area, or in order to account for possible deceptive strategies of an opponent.

In contrast to most standard pursuit-evasion problem formulations, where the effect of the environment is not taken into consideration, in our problem setup (only) the pursuers will be affected by known exogenous environmental conditions (e.g., winds or sea currents). It will also be

assumed that each pursuer has a stroboscopic view of the target position. That is, each pursuer knows the current position of the target but not its future position nor its velocity. Our objective is to find which pursuer will go after the target at each instant of time so as to reduce or minimize the capture time.

The main tool we use to find "areas of influence" of each pursuer are Zermelo-Voronoi diagrams.

The Zermelo-Voronoi Diagram

When we deal with pursuer-target problems, in many cases we want to know the proximity relation between a set of agents, acting as pursuers, and a target on the plane. The problem of obtaining this proximity relation can often be recast as a set membership problem. For instance, the question of determining which of the agents is closest (in terms of arrival time) to a static target at a particular instant of time, reduces to a set membership problem, namely, one of forming the so-called Zermelo-Voronoi Diagram (ZVD), and then finding the cell in which the target resides at the given time instant. We state the precise definition of the ZVD below.

Definition 1. *Given a set of n agents starting from distinct initial positions, whose dynamics are given by*

$$\dot{X}_P^i = u_P^i + w(X_P^i, t), \qquad X_P^i(0) = X_{P_0}^i, \tag{115}$$

where $X_P^i := [x_P^i, y_P^i]^\mathsf{T} \in \mathbb{R}^2$ *denotes the position of the* i^{th} *agent,* $u_P^i \in \mathbb{R}^2$ *is the control input of the* i^{th} *agent, and* $w(X_P^i, t)^T \in \mathbb{R}^2$ *represents the wind disturbance, the* Zermelo-Voronoi diagram *(ZVD) (or Zermelo-Voronoi partition) is a set partition of the plane* $\mathcal{Z} = \{Z_1, Z_2, ..., Z_n\}$ *such that*

i) $\mathbb{R}^2 = \bigcup_{i=1}^n Z_i$,

ii) For any point in Z_i*, the agent i will reach this point faster than any other agent.*

The sets Z_i *are the* Zermelo-Voronoi cells *for the partition.*

It turns out that in the case when the wind field is only time-varying, $w(X_P^i, t) = w(t)$, there exists a homeomorphism between the ordinary Voronoi diagram and the Zermelo-Voronoi diagram with the same generators. This allows the construction of the ZVD efficiently using known VD algorithms from computational geometry.

Problem Setup

Consider a group of n pursuers in the plane, denoted by the index set $I = \{1, 2, ..., n\}$, and assume that at time $t = 0$ the pursuers are located at n distinct positions in the plane, designated by

$\mathcal{P}_0 = \{X^i_{P_0} \in \mathbb{R}^2, i \in I\}$. The kinematics of the i^{th} pursuer, $i \in I$, are described by (115), where it is assumed that $u^i_P \in \mathcal{U}_P$, where the set $\mathcal{U}_P$ consists of all piecewise continuous functions whose range is included in the set $U_P = \{u \in \mathbb{R}^2, |u| \leqslant \bar{u}\}$. It is assumed, furthermore, that there exists $0 < \bar{w} < \bar{u}$ such that

$$|w(t)| \leqslant \bar{w}, \tag{116}$$

for all $t \geqslant 0$. The restriction on the magnitude of the wind disturbance is imposed in order to ensure complete pursuer controllability, namely, that the pursuers are able to reach any point on the plane in finite time.

The objective of the pursuers is to intercept a target, whose kinematics is given by

$$\dot{X}_T = u_T, \qquad X_T(0) = X_{T_0}, \tag{117}$$

where $X_T = [x_T, y_T]^\mathsf{T} \in \mathbb{R}^2$ is the position of the target, and $u_T \in \mathbb{R}^2$ is its control input such that $u_T \in \mathcal{U}_T$. The set $\mathcal{U}_T$ which consists of all piecewise continuous functions whose range is included in the set $U_T = \{u \in \mathbb{R}^2, |u| \leqslant \bar{q}\}$. Note that the target is not affected by the wind field.

We assume that the pursuers have accurate measurements only of the current position of the target at every instant of time. One reasonable strategy for every pursuer is therefore to use the Zermelo navigation law in order to intercept the target, that is, at every instant of time, the pursuer approaches the target with the control law obtained by the solution of the corresponding Zermelo navigation problem. This control law is optimal at $t = 0$ if the target remains stationary for all $t \geqslant 0$. Starting at time $t = 0$, the optimal time of arrival T^i_{ZN} of the i^{th} pursuer from $X^i_{P_0}$ to X_{T_0} is given by

$$T^i_{\text{ZN}} = \min\{T > 0 : \bar{u}T - |X_{T_0} - X^i_{P_0} - \int_0^T w(\tau)\,\mathrm{d}\tau| = 0\} \tag{118}$$

Then the Zermelo's navigation control can be obtained by

$$u^i_{\text{ZN}} = \bar{u}(\cos\theta^*_i, \sin\theta^*_i)^\mathsf{T}, \tag{119}$$

where

$$\theta^*_i = \text{Arg}\big(X_{T_0} - X^i_{P_0} - \int_0^{T^i_{\text{ZN}}} w(\tau)\,\mathrm{d}\tau\big), \tag{120}$$

for $i \in I$.

It can be shown that a sequential pursuit strategy in which each pursuer employs the Zermelo's navigation law leads to capture of the target by at least one pursuer.

We may now propose the following algorithm to assign the active pursuers:

Dynamic Assignment of Active Pursuer

a) Construct the ZVD and assign the i^{th} pursuer to be the active pursuer if the target resides in the corresponding Zermelo-Voronoi cell Z_i.

b) At every time step, generate the ZVD and assign the j^{th} pursuer to be the active pursuer if the target resides in the corresponding Zermelo-Voronoi cell Z_j.

c) Check the distance between the target and the active pursuer and repeat step b) if the distance is bigger than ϵ. Otherwise, terminate the procedure and return the sequence of active pursuers.

The update algorithm to dynamically generate the ZVD from one time step to the next when a single pursuer has moved makes use of the dual of the VD, namely, its Delaunay Triangulation. In order to update the ordinary VD we will, instead, update its dual graph. The algorithm for updating the Zermelo-Voronoi diagram from the previous time step to current time step is given in Algorithm 1.

Algorithm 1 Update Zermelo Voronoi Diagram

Input: Coordinates $\mathcal{P}_{k-1}$ of the generators at the previous time step and the corresponding Delaunay triangulation DT, coordinates $\mathcal{P}_k$ of point set at current time step.
Output: Updated Zermelo Voronoi Diagram and Delaunay Triangulation at the current time step.

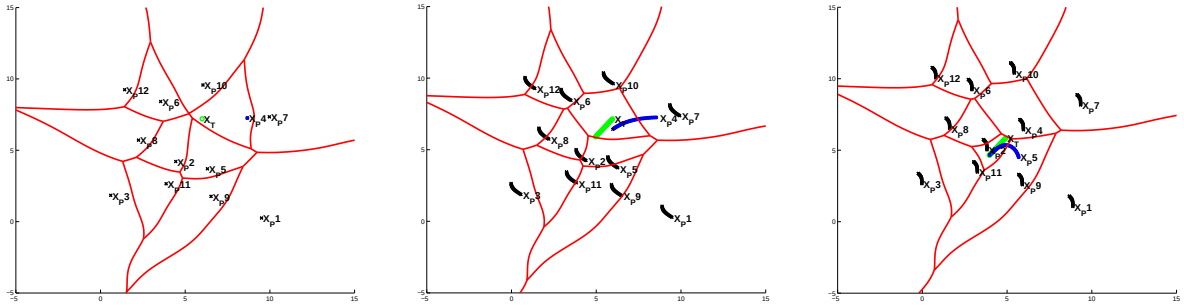
procedure UPDATE_DT(DT,P_{k-1},P_k)
  while the triangulation DT is not embedded under current coordinates $\mathcal{P}_k$ **do**
    Update DT by removing one of the points that cause the unembedding (in our case the active pursuer);
    store the current coordinates of removed points into set R;
  end while
  if R is not empty **then**
    flip the remaining triangulation into a Delaunay triangulation;
  end if
  for $i = 0$ to $length(R)$ **do**
    Update DT by inserting the i^{th} point in R into the triangulation;
  end for
  Transform DT into an ordinary Voronoi diagram VD;
  Transform VD into the ZVD at current time.
  return ZVD and DT.
end procedure

Simulation Results

Consider a scenario where the target is moving in a straight line according to equation (117), where $u_T(t) = [-0.4, -0.5]^\mathsf{T}$. Assume that there exist 12 pursuers, each having maximum unit speed ($\bar{u} = 1$), which are initially located at distinct positions determined by $\mathcal{P}_0$. The wind field that affects the pursuers is given by

$$w(t) = \begin{bmatrix} -0.2 - 0.2\cos(t) \\ 0.3 \end{bmatrix}.$$

Figures 25 illustrate the trajectories of the pursuers in the wind and the moving target. Specifically, Figure 25(a) shows the ZVD formed by the pursuers at $t = 0$. As seen in this figure, $i = 4$ is the active pursuer since the target falls in the Zermelo-Voronoi cell of X_P^4. Figure 25(b) illustrates the trajectories of the target and the pursuers in the time interval $[0, \tau_1]$, where $\tau_1 = 2.6$ is the switching time. The Zermelo-Voronoi Diagram at $t = \tau_1$ is also presented to show that the target is about to leave the Zermelo-Voronoi cell of X_P^4 and enter another cell. Figure 25(c) shows the trajectories of the target and the pursuers from $t = \tau_1$ to capture time $T_c = 5.0$, as well as the Zermelo-Voronoi Diagram at $t = T_c$. In the last time interval the target is assigned to $i = 5$.



(a) (b) (c)

Figure 25: (a) Zermelo-Voronoi Diagram formed by pursuers at $t = 0$, X_P^4 is the active pursuer, (b) Zermelo-Voronoi Diagram formed by pursuers at the first switch time $t = 2.6$, and trajectories of pursuers and target for $t \in [0, 2.6)$, (c) Zermelo-Voronoi Diagram formed by pursuers at the capture time $T_c = 5.0$, and trajectories of pursuers and target for $t \in [2.6, 5.0]$, X_P^5 is the active pursuer.

For comparison, note that when only one pursuer tries to capture the target, the shortest possible time is $T_c = 7.5$. In that case there is a single active pursuer, namely, X_P^4.

3.5 Pursuit-Evasion Games in Dynamic Flow Fields via Reachability Set Analysis

Despite the plethora of work in this area, few approaches have taken into consideration how dynamic environmental conditions may affect the outcome of the game. For instance, when either the pursuer or the evader (or both) is a small autonomous underwater vehicle (AUV) or small unmanned aerial vehicle (UAV), the presence of dynamic sea currents or winds may significantly affect the vehicle motion. As a result, during the pursuit-evasion of these vehicles, their optimal behavior, as the solution of a differential game, may be greatly affected by the existence of the external dynamic flow field.

Some optimal control problems have taken into account the effect of an external flow field. However, the same level of attention has not been shared in the literature for pursuit-evasion games under the influence of external disturbances. Here, we consider a two-player pursuit-evasion game in an external dynamic flow field. Due to the generality of the external flow, Issacs' approach does not readily yield feasible results. Instead, we find the optimal trajectories of the players through the evolution of their reachable sets. We utilize the level set method to generate the reachable sets and retrieve the corresponding optimal controls by backward propagation of their respective reachable sets.

Problem Formulation

Consider a pursuit-evasion game in an external dynamic flow field with a single pursuer P and a single evader E. The dynamics of the pursuer P is given by

$$\dot{X}_P = u_P + w(X_P, t), \qquad X_P(0) = X_{P_0}, \tag{121}$$

where $X_P = [x_P, y_P]^{\mathrm{T}} \in \mathbb{R}^2$ denotes the position of the pursuer, $u_P \in \mathbb{R}^2$ is the control input of the pursuer such that $u_P \in \mathcal{U}_P$. The set $\mathcal{U}_P$ consists of all piecewise continuous functions, whose range is included in the set $U_P = \{u \in \mathbb{R}^2, |u| \leqslant \bar{u}\}$, where $|\cdot|$ represents the 2-norm. If $u_P \in \mathcal{U}_P$, we say that u_P is an *admissible* control of the pursuer. In (121), $w(X,t) \in \mathbb{R}^2$ represents the external dynamic flow. It is reasonable to assume that the magnitude of this flow (e.g. winds or currents) is bounded from above by some constant, hence there exists a constant $\bar{w}$ such that $|w(X,t)| \leq \bar{w}$, for all (X,t). Here, we assume that the effects of the external dynamic flow field on the pursuer and evader are identical.

The objective of the pursuer is to intercept an evader, whose kinematics is given by

$$\dot{X}_E = u_E + w(X_E, t), \qquad X_E(0) = X_{E_0}, \tag{122}$$

where $X_E = [x_E, y_E]^{\mathrm{T}} \in \mathbb{R}^2$ is the position of the evader, and u_E is its control input such that $u_E \in \mathcal{U}_E$, which consists of all piecewise continuous functions whose range is included in the set $U_E = \{v \in \mathbb{R}^2, |v| \leqslant \bar{v}\}$. Again, u_E is an *admissible* control of the evader if $u_E \in \mathcal{U}_E$.

The game begins at the initial time $t_0 = 0$ with initial positions X_{E_0}, X_{P_0}, and terminates when the pursuer reaches the location of the evader, that is, there exists a terminal time T such that $X_P(T) = X_E(T)$. Specifically, the game terminates if there exists an admissible control $u_P \in \mathcal{U}_P$ of the pursuer such that $X_P(T) = X_E(T)$ for some finite time $T > 0$, regardless of any admissible control of the evader $u_E \in \mathcal{U}_E$.

Problem Analysis

Reachable Sets. A reachable (or attainable) set at a given time is defined as the set of points that can be visited by the agent at a particular time. Note that, in our problem, the pursuer can pick its speed within the range of $[0, \bar{u}]$. When the maximum external flow magnitude is smaller than the pursuer's speed, the set of reachable points at a given time is identical to the set of points the pursuer can reach *within* that time. If the external flow is at some time and locations larger than the speed of the pursuer, the reachable set is the set of points the pursuer can reach *at* that time. The same holds for the evader. In all cases, the boundary of the reachable set is called the *reachability front*. In particular, the reachable set of the pursuer at time $t \geq 0$, denoted by $\mathcal{R}_P(X_{P_0}, t)$, is the set of all points $X \in \mathbb{R}^2$ such that there exists a trajectory satisfying (115), with initial position X_{P_0} and terminal position X at time t. Similarly, the reachable set $\mathcal{R}_E(X_{E_0}, t)$ of the evader at time $t \geq 0$ is the set of all points $Y \in \mathbb{R}^2$ such that there exists a trajectory satisfying (122), with initial position X_{E_0} and terminal position Y at time t. The reachability fronts of the pursuer and the evader at time $t \geq 0$ are denoted by $\partial\mathcal{R}_P(X_{P_0}, t)$ and $\partial\mathcal{R}_E(X_{E_0}, t)$, respectively. We denote by $\mathcal{R}_E^\star(X_{E_0}, t)$ the *usable reachable set* of the evader, which is the set of all terminal points (at time t) of admissible trajectories of the evader that do not pass through the reachable set of the pursuer at *any* time in the interval $[0, t]$. In other words, $\mathcal{R}_E^\star(X_{E_0}, t)$ is the set of terminal points of all the 'safe' evader trajectories.

Proposition 3. *Let $T = \inf\{t \in \mathbb{R} : \mathcal{R}_E^\star(X_{E_0}, t) = \emptyset\}$. If $T < \infty$, then capture is guaranteed for any time greater than T, while the evader can always escape within a time smaller than T. That is, T is the optimal time-to-capture. Moreover, let X_f denote the location where the evader is captured by the pursuer. Then X_f lies on the intersection of the reachability front of the pursuer $\partial\mathcal{R}_P(X_{P_0}, T)$ and the reachability front of the evader $\partial\mathcal{R}_E(X_{E_0}, T)$.*

Remark 2. *In most cases, such as when $\bar{u} \geq \bar{v}$, the safe reachable set of the evader $\mathcal{R}_E^\star(X_{E_0}, t)$ satisfies*

$$\mathcal{R}_E^\star(X_{E_0}, t) = \mathcal{R}_E(X_{E_0}, t) \backslash \mathcal{R}_P(X_{P_0}, t), \tag{123}$$

for all $t \geq 0$. In such cases, the condition $\mathcal{R}_E^\star(X_{E_0}, t) = \emptyset$ is equivalent to the condition $\mathcal{R}_E(X_{E_0}, t) \subseteq \mathcal{R}_P(X_{P_0}, t)$. Then, the optimal time-to-capture is the first time when the reachable set of the pursuer $\mathcal{R}_P(X_{P_0}, t)$ completely covers the reachable set of the evader $\mathcal{R}_E(X_{E_0}, t)$. When $\bar{u} < \bar{v}$ (the relative maximum speed of the evader is larger than the maximum speed of the evader) the relation (123) may not always hold–some admissible evader trajectories may temporarily enter the reachable set of the pursuer and exit later. In these cases, $\mathcal{R}_E^\star(X_{E_0}, t)$ can be determined by treating the reachable

set of the pursuer as a dynamic 'forbidden' region for the evader. However, the examples shown in this paper use $\bar{u} \geq \bar{v}$, which allows us to exploit (123).

Proposition 3 is valid under the assumption that capture is guaranteed for some finite time. We provide a sufficient condition for this to occur in the next theorem.

Theorem 3. *Assume $w(X,t)$ satisfies the triangle inequality and its norm is bounded from above by a constant λ, where $\lambda < \bar{u} - \bar{v}$. Then the game terminates in finite time regardless of the initial positions between the pursuer and the evader. Furthermore, the time-to-capture satisfies the upper bound*

$$T \leq \frac{|X_{E_0} - X_{P_0}|}{\bar{u} - \bar{v} - \lambda}. \tag{124}$$

Level Set Method. In order to construct the forward reachable sets of the pursuers and the evader, we utilize the level set method. The level set method is a convenient tool to track the evolution of the reachability front. It evolves an interface (front) by embedding it as a hypersurface in a higher dimension, where time is the augmented dimension. Automatic handling of merging and pinching of fronts and other topological changes are made possible by such higher dimensional embedding. Level sets provide an implicit representation of the front, which offers several advantages over an explicit representation.

For any $c \in \mathbb{R}$, the c-level set of a function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is the set $\{X \in \mathbb{R}^2 | \phi(X) = c\}$. We consider the *signed distance function*

$$\phi(X) = \begin{cases} \min\limits_{Y \in \partial\mathcal{R}} |X - Y|, & \text{if } X \text{ is outside the front,} \\ -\min\limits_{Y \in \partial\mathcal{R}} |X - Y|, & \text{if } X \text{ is inside the front.} \end{cases} \tag{125}$$

The signed distance function is one of the most commonly used implicit functions in level sets. It is smooth and monotonic across the interface. It also keeps fixed amplitude gradients in the field. For all $X \in \partial\mathcal{R}$, we have $\phi(X) = 0$. That is, the zero level set implicitly represents the reachability front. Moreover, the reachable set can be represented by $\{X \in \mathbb{R}^2 | \phi(X) \leq 0\}$.

The reachability front $\partial\mathcal{R}_P(X_{P_0}, t)$ of the pursuer is governed by the viscosity solution of the Hamilton-Jacobi (HJ) equation

$$\frac{\partial \phi_P(X,t)}{\partial t} + \bar{u} \, |\nabla \phi_P| + w(X,t) \cdot \nabla \phi_P = 0, \tag{126}$$

with initial conditions $\phi_P(X,0) = |X - X_{P_0}|$. Moreover, the reachable set of the pursuer coincides with the region(s) where ϕ_P is non-positive. Similarly, the reachability front $\partial\mathcal{R}_E(X_{E_0}, t)$ of the evader is given by the HJ equation

$$\frac{\partial \phi_E(X,t)}{\partial t} + \bar{v} \, |\nabla \phi_E| + w(X,t) \cdot \nabla \phi_E = 0, \tag{127}$$

with initial conditions $\phi_E(X,0) = |X - X_{E_0}|$.

In the case when $\bar{v} > \bar{u}$, we need to track the propagation of $\partial\mathcal{R}_E^\star(X_{E_0},t)$. Its reachability front can be computed by solving the following modified version of the Hamilton-Jacobi equation (127):

$$\frac{\partial\phi_E^\star(X,t)}{\partial t} + \tilde{v}(t)\,|\nabla\phi_E^\star| + w(X,t)\cdot\nabla\phi_E^\star = 0, \tag{128}$$

where

$$\tilde{v}(t) = \begin{cases} \bar{v}, \text{ if } \phi_P(X,t) \geq 0, \\ \bar{u}, \text{ otherwise.} \end{cases} \tag{129}$$

The main idea is to propagate $\mathcal{R}_E^\star(X_{E_0},t)$ with the maximum speed of the evader $\bar{v}$ when it is outside the reachable set of the pursuer, and to keep pace with the propagation of $\partial\mathcal{R}_P(X_{P_0},t)$ when the front of the evader enters the reachable set of the pursuer to make sure that it never grows out of the reachable set of the pursuer again. Note that $\mathcal{R}_E^\star(X_{E_0},t)$ is represented by the area $\{X \in \mathbb{R}^2 | \phi_E^\star(X,t) \leq 0 \text{ and } \phi_P(X,t) \geq 0\}$.

Time-Optimal Paths.

The location X_f where the evader is captured by the pursuer is the intersection between $\partial\mathcal{R}_P(X_{P_0},T)$ and $\partial\mathcal{R}_E(X_{E_0},T)$. After we have identified the (common) terminal position of the pursuer and the evader, we can retrieve the optimal trajectories and optimal controls of both players by backward propagation along the reachable sets. In particular, the time-optimal trajectories X_P^* and X_E^* satisfy the following differential equations, when ϕ_P and ϕ_E are differentiable:

$$\frac{\mathrm{d}X_P^*}{\mathrm{d}t} = \bar{u}\frac{\nabla\phi_P}{|\nabla\phi_P|} + w(X_P^*,t), \tag{130}$$

$$\frac{\mathrm{d}X_E^*}{\mathrm{d}t} = \bar{v}\frac{\nabla\phi_E}{|\nabla\phi_E|} + w(X_E^*,t). \tag{131}$$

Hence, the corresponding time-optimal controls of the pursuer and the evader are

$$u_P^* = \bar{u}\frac{\nabla\phi_P}{|\nabla\phi_P|}, \qquad u_E^* = \bar{v}\frac{\nabla\phi_E}{|\nabla\phi_E|}. \tag{132}$$

Simulation Results

In this section, we apply our method to a problem with a more realistic representation of the flow field. In particular, we consider a wind field approximation generalized from the Rankine model of vortex:

$$w(X) = w_0 + \sum_{i=1}^{n_s} \omega_i A_i (X - x_{s_i}), \tag{133}$$

where

$$\omega_i = \frac{1}{\max\{r_{s_i}^2, \|X - x_{s_i}\|^2\}}. \tag{134}$$

In (133) n_s is the number of flow singularities, x_{s_i} is the location of the i-th flow singularity and r_{s_i} denotes the singularity radius. A_i is a 2×2 matrix, whose structure captures the local characteristics of the i-th flow singularity. The model approximates the velocity field of a vortex with a linear vector field inside a disk and the velocity outside of the disk decreases as the inverse squared distance to the center of the disk.

For our numerical simulation, we set the number of flow singularities to $n_s = 3$. The locations of the flow singularities are $x_{s_1} = [3, 2]^{\mathrm{T}}$, $x_{s_2} = [-3, 4]^{\mathrm{T}}$, $x_{s_3} = [0, -5]^{\mathrm{T}}$, and the corresponding radii are $r_{s_1} = 3$, $r_{s_2} = 2$, $r_{s_3} = 3$, respectively. The local wind field matrices are given by

$$A_1 = \begin{bmatrix} 0 & 0.3 \\ -0.15 & 0 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0.4 & 0.2 \\ 0 & -0.2 \end{bmatrix}, \qquad A_3 = \begin{bmatrix} 0.2 & 0.1 \\ -0.2 & 0.2 \end{bmatrix}.$$

We also choose $w_0 = [0, 0]$.

The reachable fronts of the pursuer and the evader at the terminal time are shown in Figure 26(a). The corresponding optimal trajectories of the pursuer and the evader are shown in Figure 26(b). Expansion of the level sets and the backward propagation step to find the optimal paths are depicted in Figures 27(a) and 27(b) for the pursuer and the evader, respectively.

3.6 Min-Max Differential Dynamic Programming: Continuous and Discrete Time Formulations

Differential game theoretic or min-max formulations are important extensions of optimal control having direct connections to robust and H^∞ nonlinear control theory. While there has been a lot of work on DDP algorithms and applications, most of the work is on discrete time formulations and for cases where there are no disturbances. Given all this existing work , our contribution in this paper is the derivation of min-max DDP in continuous time and its comparison with the discrete time formulation. In particular, we provide a set of backward differential equations for both continuous and discrete time that are easy to implement and derive the optimal policies for the two players/controllers. We compare the continuous and discrete time formulations in terms of convergence and numerical efficiency. In addition we investigate the effect that the min-max formulation has in the feed-forward and feedback parts of the optimal control policies.

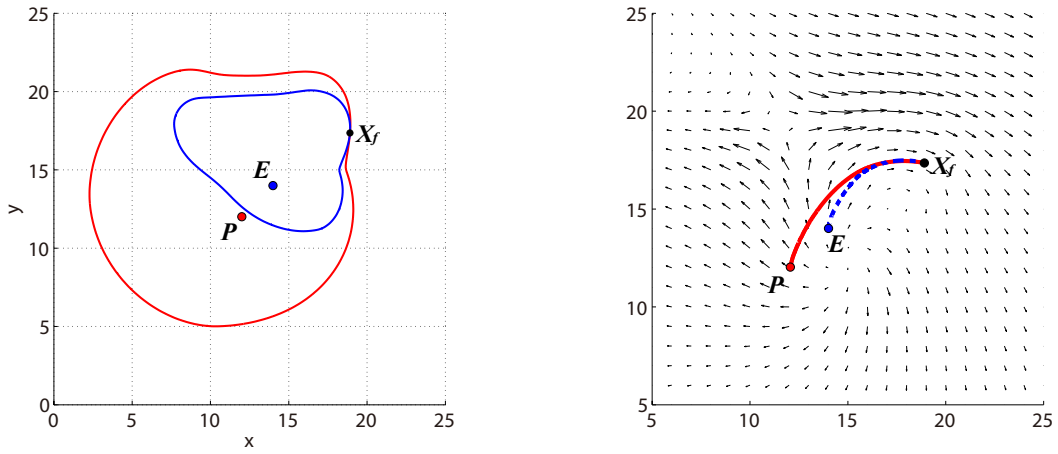


Figure 26: (a) Red and blue curves represent the reachable fronts of the pursuer and the evader at the terminal time, respectively. They intersect at X_f, where the pursuer captures the evader eventually. (b) Optimal trajectories of the pursuer and the evader in red and dotted blue lines, respectively, generated from the reachable set approach. The wind field is depicted in the background.

Game Theoretic Differential Dynamic Programming in Continuous Time

On the differential game theoretic setting, we have the following min-max problem:

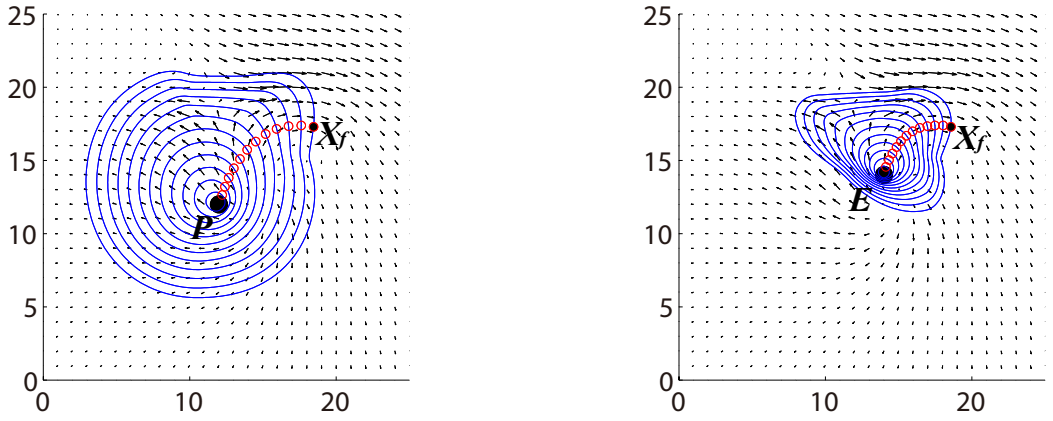
$$V(\mathbf{x}(t_0), t_0) = \min_{\mathbf{u}} \max_{\mathbf{v}} \left\{ \phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}, t) \mathrm{d}t \right\}, \tag{135}$$

subject to the dynamics

$$\mathrm{d}\mathbf{x} = F(\mathbf{x}, \mathbf{u}, \mathbf{v}, t)\mathrm{d}t, \tag{136}$$

where V stands for the performance index, $\mathbf{x}(t)$ is an n-dimensional vector function of time describing the state of the dynamic system at $t \in [0, t_f]$. $\mathcal{L}$ and ϕ are scalar functions of their arguments, where $\mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{v}, t)$ is the *running cost* and $\phi(\mathbf{x}(t_f), t_f)$ is the *terminal cost*. $\mathbf{u}$ is an m-dimensional vector function that represents the stabilizing control of the system, whose objective is to minimize the performance index. Whereas $\mathbf{v}$ being a q-dimensional vector represents the destabilizing control of the system, that tries to maximize the performance index.

In continuous time, the analysis starts with the Hamilton-Jacobi-Bellman Isaacs (HJBI) partial differential equation. More precisely, we have:



(a) (b)

Figure 27: (a) Optimal trajectory of the pursuer in red dots overlaid on intermediate reachability front contours in blue. (b) Optimal trajectory of the evader in red dots overlaid on intermediate reachability front contours in blue.

$$-\frac{\partial V(\mathbf{x},t)}{\partial t} = \min_{\mathbf{u}} \max_{\mathbf{v}} \left\{ \mathcal{L}(\mathbf{x},\mathbf{u},\mathbf{v},t) + V_{\mathbf{x}}(\mathbf{x},t)^{\mathrm{T}} F(\mathbf{x},\mathbf{u},\mathbf{v},t) \right\}, \tag{137}$$

with boundary condition

$$V(\mathbf{x},t_f) = \phi(\mathbf{x}(t_f),t_f). \tag{138}$$

Given an initial/nominal trajectory of the state and control $(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{v}})$, and letting $\delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$, $\delta\mathbf{u} = \mathbf{u} - \bar{\mathbf{u}}$, $\delta\mathbf{v} = \mathbf{v} - \bar{\mathbf{v}}$, the linearized dynamics can be represented as

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \bar{F}(\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}, \bar{\mathbf{v}} + \delta\mathbf{v}, t), \tag{139}$$

$$\frac{\mathrm{d}\delta\mathbf{x}}{\mathrm{d}t} = \bar{F}_{\mathbf{x}}\delta\mathbf{x} + \bar{F}_{\mathbf{u}}\delta\mathbf{u} + \bar{F}_{\mathbf{v}}\delta\mathbf{v}, \tag{140}$$

where $bF_{\mathbf{x}}$, $\bar{F}_{\mathbf{u}}$, and $\bar{F}_{\mathbf{v}}$ stand for $F_{\mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{v}},t)$, $F_{\mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{v}},t)$, $F_{\mathbf{v}}(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{v}},t)$, respectively. Henceforth, the arguments for the functions V, F, etc, are omitted for brevity unless otherwise specified, and a bar $\bar{\cdot}$ is attached when they are evaluated at the nominal trajectory $(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{v}})$.

The main idea here is to take expansions of the terms in both sides of the equation (137) around the nominal state and control trajectories $(\bar{\mathbf{x}},\bar{\mathbf{u}},\bar{\mathbf{v}})$ to derive the update law for the stabilizing control, destabilizing control and backward differential equations for the zeroth, first and second

order approximation terms of the value function. After domr mathematical manipulations, we obtain

$$\delta \mathbf{u}^* = -Q_{\mathbf{uu}}^{-1}(Q_{\mathbf{ux}}\delta \mathbf{x} + Q_{\mathbf{uv}}\delta \mathbf{v} + Q_{\mathbf{u}}), \tag{141}$$

$$\delta \mathbf{v}^* = -Q_{\mathbf{vv}}^{-1}(Q_{\mathbf{vx}}\delta \mathbf{x} + Q_{\mathbf{vu}}\delta \mathbf{u} + Q_{\mathbf{v}}), \tag{142}$$

where $Q_{\mathbf{vu}} = Q_{\mathbf{uv}}^{\mathrm{T}}$, and

$$\begin{aligned}
&Q_{\mathbf{x}} = \bar{F}_{\mathbf{x}}^{\mathrm{T}}\bar{V}_{\mathbf{x}} + \bar{\mathcal{L}}_{\mathbf{x}}, && Q_{\mathbf{u}} = \bar{F}_{\mathbf{u}}^{\mathrm{T}}\bar{V}_{\mathbf{x}} + \bar{\mathcal{L}}_{\mathbf{u}}, && Q_{\mathbf{v}} = \bar{F}_{\mathbf{v}}^{\mathrm{T}}\bar{V}_{\mathbf{x}} + \bar{\mathcal{L}}_{\mathbf{v}},\\
&Q_{\mathbf{xx}} = \bar{\mathcal{L}}_{\mathbf{xx}} + 2\bar{V}_{\mathbf{xx}}\bar{F}_{\mathbf{x}}, && Q_{\mathbf{uu}} = \bar{\mathcal{L}}_{\mathbf{uu}}, && Q_{\mathbf{vv}} = \bar{\mathcal{L}}_{\mathbf{vv}},\\
&Q_{\mathbf{ux}} = \bar{F}_{\mathbf{u}}^{\mathrm{T}}\bar{V}_{\mathbf{xx}} + \bar{\mathcal{L}}_{\mathbf{ux}}, && Q_{\mathbf{vx}} = \bar{F}_{\mathbf{v}}^{\mathrm{T}}\bar{V}_{\mathbf{xx}} + \bar{\mathcal{L}}_{\mathbf{vx}}, && Q_{\mathbf{uv}} = \bar{\mathcal{L}}_{\mathbf{uv}}.
\end{aligned} \tag{143}$$

Notice that $\delta \mathbf{v}$ is still in the previous expression of $\delta \mathbf{u}^*$. We need to replace the $\delta \mathbf{v}$ term in (141) with (142) and solve for $\delta \mathbf{u}^*$. Similarly, we can solve for $\delta \mathbf{v}^*$. The final expressions for $\delta \mathbf{u}^*$ and $\delta \mathbf{v}^*$ are specified as follows:

$$\delta \mathbf{u}^* = \mathbf{l}_{\mathbf{u}} + \mathbf{K}_{\mathbf{u}}\delta \mathbf{x} \quad \text{and} \quad \delta \mathbf{v}^* = \mathbf{l}_{\mathbf{v}} + \mathbf{K}_{\mathbf{v}}\delta \mathbf{x}, \tag{144}$$

with the feed-forward gains $\mathbf{l}_{\mathbf{v}}, \mathbf{l}_{\mathbf{u}}$ and feedback gains $\mathbf{K}_{\mathbf{v}}, \mathbf{K}_{\mathbf{u}}$ defined as:

$$\mathbf{l}_{\mathbf{u}} = -(Q_{\mathbf{uu}} - Q_{\mathbf{uv}}Q_{\mathbf{vv}}^{-1}Q_{\mathbf{vu}})^{-1}(Q_{\mathbf{u}} - Q_{\mathbf{uv}}Q_{\mathbf{vv}}^{-1}Q_{\mathbf{v}}), \tag{145}$$

$$\mathbf{l}_{\mathbf{v}} = -(Q_{\mathbf{vv}} - Q_{\mathbf{vu}}Q_{\mathbf{uu}}^{-1}Q_{\mathbf{uv}})^{-1}(Q_{\mathbf{v}} - Q_{\mathbf{vu}}Q_{\mathbf{uu}}^{-1}Q_{\mathbf{u}}), \tag{146}$$

$$\mathbf{K}_{\mathbf{u}} = -(Q_{\mathbf{uu}} - Q_{\mathbf{uv}}Q_{\mathbf{vv}}^{-1}Q_{\mathbf{vu}})^{-1}(Q_{\mathbf{ux}} - Q_{\mathbf{uv}}Q_{\mathbf{vv}}^{-1}Q_{\mathbf{vx}}), \tag{147}$$

$$\mathbf{K}_{\mathbf{v}} = -(Q_{\mathbf{vv}} - Q_{\mathbf{vu}}Q_{\mathbf{uu}}^{-1}Q_{\mathbf{uv}})^{-1}(Q_{\mathbf{vx}} - Q_{\mathbf{vu}}Q_{\mathbf{uu}}^{-1}Q_{\mathbf{ux}}). \tag{148}$$

The next step is to substitute the optimal control (141) and disturbance (destabilizing control) (142) to the HJBI equation (137) in order to find the update law of the value function and its first and second order partial derivatives. After collecting terms as zeroth order, first order and second order expressions of $\delta \mathbf{x}$, we can equate the coefficients of $\delta \mathbf{x}$ and readily obtain the backward propagation equations with respect to the value function and its first and second order partial derivatives. These backward differential equations are expressed as follows

$$\begin{aligned}
-\frac{\mathrm{d}\bar{V}}{\mathrm{d}t} &= \bar{\mathcal{L}} + \mathbf{l}_{\mathbf{u}}^{\mathrm{T}}Q_{\mathbf{u}} + \mathbf{l}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{v}} + \frac{1}{2}\mathbf{l}_{\mathbf{u}}Q_{\mathbf{uu}}\mathbf{l}_{\mathbf{u}} + \mathbf{l}_{\mathbf{u}}^{\mathrm{T}}Q_{\mathbf{uv}}\mathbf{l}_{\mathbf{v}} + \frac{1}{2}\mathbf{l}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{vv}}\mathbf{l}_{\mathbf{v}},\\
-\frac{\mathrm{d}\bar{V}_{\mathbf{x}}}{\mathrm{d}t} &= Q_{\mathbf{x}} + \mathbf{K}_{\mathbf{u}}^{\mathrm{T}}Q_{\mathbf{u}} + \mathbf{K}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{v}} + Q_{\mathbf{ux}}^{\mathrm{T}}\mathbf{l}_{\mathbf{u}} + Q_{\mathbf{vx}}^{\mathrm{T}}\mathbf{l}_{\mathbf{v}} + \mathbf{K}_{\mathbf{u}}^{\mathrm{T}}Q_{\mathbf{uu}}\mathbf{l}_{\mathbf{u}} + \mathbf{K}_{\mathbf{u}}^{\mathrm{T}}Q_{\mathbf{uv}}\mathbf{l}_{\mathbf{v}}\\
&\quad + \mathbf{K}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{vu}}\mathbf{l}_{\mathbf{u}} + \mathbf{K}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{vv}}\mathbf{l}_{\mathbf{v}},\\
-\frac{\mathrm{d}\bar{V}_{\mathbf{xx}}}{\mathrm{d}t} &= 2\mathbf{K}_{\mathbf{u}}^{\mathrm{T}}Q_{\mathbf{ux}} + 2\mathbf{K}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{vx}} + 2\mathbf{K}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{vu}}\mathbf{K}_{\mathbf{u}} + \mathbf{K}_{\mathbf{u}}^{\mathrm{T}}Q_{\mathbf{uu}}\mathbf{K}_{\mathbf{u}} + \mathbf{K}_{\mathbf{v}}^{\mathrm{T}}Q_{\mathbf{vv}}\mathbf{K}_{\mathbf{v}} + Q_{\mathbf{xx}}.
\end{aligned} \tag{149}$$

Game Theoretic Differential Dynamic Programming in Discrete Time

In the discrete-time approach, the problem is discretized along the time interval. Similar to the HJBI equation in continuous-time case, our analysis in discrete time starts with a variation of the Bellman equation for the min-max case with $\mathbf{u}(t_k)$ and $\mathbf{v}(t_k)$:

$$V(\mathbf{x}(t_k), t_k) = \min_{\mathbf{u}(t_k)} \max_{\mathbf{v}(t_k)} \Big\{ L(\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{v}(t_k), t_k) + V(\mathbf{x}(t_{k+1}), t_{k+1}) \Big\}. \tag{150}$$

Consider again the dynamical system $\mathrm{d}\mathbf{x} = F(\mathbf{x}, \mathbf{u}, \mathbf{v})\mathrm{d}t$. The linearized dynamics model around $\bar{\mathbf{x}}(t_k), \bar{\mathbf{u}}(t_k), \bar{\mathbf{v}}(t_k)$ in discrete time can be written as $\delta\mathbf{x}(t_{k+1}) = \mathbf{A}(t_k)\delta\mathbf{x}(t_k) + \mathbf{B}_{\mathbf{u}}(t_k)\delta\mathbf{u}(t_k) + \mathbf{B}_{\mathbf{v}}(t_k)\delta\mathbf{v}(t_k)$, where $\mathbf{A}(t_k) = \mathbf{I} + \mathbf{f}_{\mathbf{x}}(t_k)\mathrm{d}t$, $\mathbf{B}_{\mathbf{u}}(t_k) = \mathbf{f}_{\mathbf{u}}(t_k)\mathrm{d}t$ and $\mathbf{B}_{\mathbf{v}}(t_k) = \mathbf{f}_{\mathbf{v}}(t_k)\mathrm{d}t$. $\delta\mathbf{x}(t_k) = \mathbf{x}(t_k) - \bar{\mathbf{x}}(t_k), \delta\mathbf{u}(t_k) = \mathbf{u}(t_k) - \bar{\mathbf{u}}(t_k)$ and $\delta\mathbf{v}(t_k) = \mathbf{v}(t_k) - \bar{\mathbf{v}}(t_k)$ are defined as the deviations from the nominal trajectory.

Let $\mathsf{Q}(\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{v}(t_k)) = L(\mathbf{x}(t_k), \mathbf{u}(t_k), \mathbf{v}(t_k), t_k) + V(\mathbf{x}(t_{k+1}), t_{k+1})$. We obtain

$$\begin{aligned} \delta\mathbf{u}(t_k)^* &= -\mathsf{Q}_{\mathbf{uu}}^{-1}\big(\mathsf{Q}_{\mathbf{ux}}\delta\mathbf{x}(t_k) + \mathsf{Q}_{\mathbf{uv}}\delta\mathbf{v}(t_k) + \mathsf{Q}_{\mathbf{u}}\big), \\ \delta\mathbf{v}(t_k)^* &= -\mathsf{Q}_{\mathbf{vv}}^{-1}\big(\mathsf{Q}_{\mathbf{vx}}\delta\mathbf{x}(t_k) + \mathsf{Q}_{\mathbf{vu}}\delta\mathbf{u}(t_k) + \mathsf{Q}_{\mathbf{v}}\big), \end{aligned} \tag{151}$$

where $\mathsf{Q}_{\mathbf{vu}} = \mathsf{Q}_{\mathbf{uv}}^{\mathrm{T}}$, and

$$\begin{aligned} \mathsf{Q}_{\mathbf{x}} = V_{\mathbf{x}}(t_{k+1})^{\mathrm{T}}\mathbf{A}(t_k) + L_{\mathbf{x}}(t_k), \quad \mathsf{Q}_{\mathbf{u}} = V_{\mathbf{x}}(t_{k+1})^{\mathrm{T}}\mathbf{B}_{\mathbf{u}}(t_k) + L_{\mathbf{u}}(t_k), \\ \mathsf{Q}_{\mathbf{v}} = V_{\mathbf{x}}(t_{k+1})^{\mathrm{T}}\mathbf{B}_{\mathbf{v}}(t_k) + L_{\mathbf{v}}(t_k), \\ \mathsf{Q}_{\mathbf{xx}} = \mathbf{A}(t_k)^{\mathrm{T}}V_{\mathbf{xx}}(t_{k+1})\mathbf{A}(t_k) + L_{\mathbf{xx}}(t_k), \mathsf{Q}_{\mathbf{uu}} = \mathbf{B}_{\mathbf{u}}(t_k)^{\mathrm{T}}V_{\mathbf{xx}}(t_{k+1})\mathbf{B}_{\mathbf{u}}(t_k) + L_{\mathbf{uu}}(t_k), \\ \mathsf{Q}_{\mathbf{vv}} = \mathbf{B}_{\mathbf{v}}(t_k)^{\mathrm{T}}V_{\mathbf{xx}}(t_{k+1})\mathbf{B}_{\mathbf{v}}(t_k) + L_{\mathbf{vv}}(t_k), \\ \mathsf{Q}_{\mathbf{xu}} = \mathbf{A}(t_k)^{\mathrm{T}}V_{\mathbf{xx}}(t_{k+1})\mathbf{B}_{\mathbf{u}}(t_k) + L_{\mathbf{xu}}(t_k), \mathsf{Q}_{\mathbf{xv}} = \mathbf{A}(t_k)^{\mathrm{T}}V_{\mathbf{xx}}(t_{k+1})\mathbf{B}_{\mathbf{v}}(t_k) + L_{\mathbf{xv}}(t_k), \\ \mathsf{Q}_{\mathbf{uv}} = \mathbf{B}_{\mathbf{u}}(t_k)^{\mathrm{T}}V_{\mathbf{xx}}(t_{k+1})\mathbf{B}_{\mathbf{v}}(t_k) + L_{\mathbf{uv}}(t_k). \end{aligned} \tag{152}$$

Solving the system of equations in (151) results in the expressions

$$\delta\mathbf{u}(t_k)^* = \bar{\mathsf{l}}_{\mathbf{u}} + \bar{\mathsf{K}}_{\mathbf{u}}\delta\mathbf{x}(t_k), \quad \delta\mathbf{v}(t_k)^* = \bar{\mathsf{l}}_{\mathbf{v}} + \bar{\mathsf{K}}_{\mathbf{v}}\delta\mathbf{x}(t_k), \tag{153}$$

where

$$\bar{\mathsf{l}}_{\mathbf{u}} = -(\mathsf{Q}_{\mathbf{uu}} - \mathsf{Q}_{\mathbf{uv}}\mathsf{Q}_{\mathbf{vv}}^{-1}\mathsf{Q}_{\mathbf{vu}})^{-1}(\mathsf{Q}_{\mathbf{u}} - \mathsf{Q}_{\mathbf{uv}}\mathsf{Q}_{\mathbf{vv}}^{-1}\mathsf{Q}_{\mathbf{v}}), \tag{154}$$

$$\bar{\mathsf{l}}_{\mathbf{v}} = -(\mathsf{Q}_{\mathbf{vv}} - \mathsf{Q}_{\mathbf{vu}}\mathsf{Q}_{\mathbf{uu}}^{-1}\mathsf{Q}_{\mathbf{uv}})^{-1}(\mathsf{Q}_{\mathbf{v}} - \mathsf{Q}_{\mathbf{vu}}\mathsf{Q}_{\mathbf{uu}}^{-1}\mathsf{Q}_{\mathbf{u}}), \tag{155}$$

$$\bar{\mathsf{K}}_{\mathbf{u}} = -(\mathsf{Q}_{\mathbf{uu}} - \mathsf{Q}_{\mathbf{uv}}\mathsf{Q}_{\mathbf{vv}}^{-1}\mathsf{Q}_{\mathbf{vu}})^{-1}(\mathsf{Q}_{\mathbf{ux}} - \mathsf{Q}_{\mathbf{uv}}\mathsf{Q}_{\mathbf{vv}}^{-1}\mathsf{Q}_{\mathbf{vx}}), \tag{156}$$

$$\bar{\mathsf{K}}_{\mathbf{v}} = -(\mathsf{Q}_{\mathbf{vv}} - \mathsf{Q}_{\mathbf{vu}}\mathsf{Q}_{\mathbf{uu}}^{-1}\mathsf{Q}_{\mathbf{uv}})^{-1}(\mathsf{Q}_{\mathbf{vx}} - \mathsf{Q}_{\mathbf{vu}}\mathsf{Q}_{\mathbf{uu}}^{-1}\mathsf{Q}_{\mathbf{ux}}). \tag{157}$$

By plugging the optimal control updates $\delta\mathbf{u}(t_k)^*$ and $\delta\mathbf{v}(t_k)^*$ into the value function, we can split the value function into zero, first and second order terms in $\delta\mathbf{x}(t_k)$ such that $V(\mathbf{x}(t_k)+\delta\mathbf{x}(t_k)) = V(t_k) + V_{\mathbf{x}}(t_k)^{\mathrm{T}}\delta\mathbf{x}(t_k) + \frac{1}{2}\delta\mathbf{x}(t_k)^{\mathrm{T}}V_{\mathbf{xx}}(t_k)\delta\mathbf{x}(t_k)$, where $V(t_k)$, $V_{\mathbf{x}}(t_k)$ and $V_{\mathbf{xx}}(t_k)$ are computed as

$$\begin{aligned}
V(t_k) &= V(t_{k+1}) + \bar{\mathsf{l}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{u}} + \bar{\mathsf{l}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{v}} + \frac{1}{2}(\bar{\mathsf{l}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{uu}}\bar{\mathsf{l}}_{\mathbf{u}} + \bar{\mathsf{l}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{vv}}\bar{\mathsf{l}}_{\mathbf{v}} + \bar{\mathsf{l}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{uu}}\bar{\mathsf{l}}_{\mathbf{v}} + \bar{\mathsf{l}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{vu}}\bar{\mathsf{l}}_{\mathbf{u}}),\\
V_{\mathbf{x}}(t_k) &= \mathsf{Q}_{\mathbf{x}} + \bar{\mathsf{K}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{u}} + \bar{\mathsf{K}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{v}} + \mathsf{Q}_{\mathbf{xu}}\bar{\mathsf{l}}_{\mathbf{u}} + \mathsf{Q}_{\mathbf{xv}}\bar{\mathsf{l}}_{\mathbf{v}} + \bar{\mathsf{K}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{uu}}\bar{\mathsf{l}}_{\mathbf{u}} + \bar{\mathsf{K}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{vv}}\bar{\mathsf{l}}_{\mathbf{v}} + \bar{\mathsf{K}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{uv}}\bar{\mathsf{l}}_{\mathbf{v}} + \bar{\mathsf{K}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{vu}}\bar{\mathsf{l}}_{\mathbf{u}},\\
V_{\mathbf{xx}}(t_k) &= \mathsf{Q}_{\mathbf{xx}} + \bar{\mathsf{K}}_{\mathbf{ux}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{ux}} + \bar{\mathsf{K}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{v}} + \mathsf{Q}_{\mathbf{xu}}\bar{\mathsf{l}}_{\mathbf{u}} + \mathsf{Q}_{\mathbf{xv}}\bar{\mathsf{l}}_{\mathbf{v}} + \bar{\mathsf{K}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{uu}}\bar{\mathsf{K}}_{\mathbf{u}} + \bar{\mathsf{K}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{vv}}\bar{\mathsf{K}}_{\mathbf{v}} + \bar{\mathsf{K}}_{\mathbf{u}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{uv}}\bar{\mathsf{K}}_{\mathbf{v}} + \bar{\mathsf{K}}_{\mathbf{v}}^{\mathrm{T}}\mathsf{Q}_{\mathbf{vu}}\bar{\mathsf{K}}_{\mathbf{u}}.
\end{aligned} \tag{158}$$

Since the optimal cost-to-go is computed backward in time, this computational scheme is called the backward-sweep in trajectory optimization.

The boundary conditions at $t = t_f$ for the backward propagations are

$$\begin{aligned}
V(t_f) &= \phi(\bar{\mathbf{x}}(t_f), t_f),\\
V_{\mathbf{x}}(t_f) &= \phi_{\mathbf{x}}(\bar{\mathbf{x}}(t_f), t_f),\\
V_{\mathbf{xx}}(t_f) &= \phi_{\mathbf{xx}}(\bar{\mathbf{x}}(t_f), t_f).
\end{aligned} \tag{159}$$

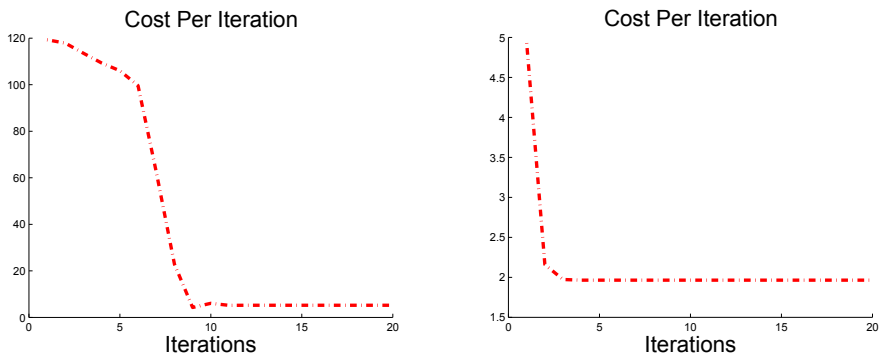
Comparison between Continuous and Discrete GT-DDP

Besides the form of the backward differential equations, one of the major differences between the discrete and continuous time formulations is on the specification of the terms $Q_{\mathbf{uu}}$ and $Q_{\mathbf{vv}}$. In the continuous case these terms are specified by $\bar{\mathcal{L}}_{\mathbf{uu}}$ and $\bar{\mathcal{L}}_{\mathbf{vv}}$ and therefore they are completely specified by the user. This is not the case with the discrete time formulation of min-max DDP in which the terms $Q_{\mathbf{uu}}$ and $Q_{\mathbf{vv}}$ are also functions of $V_{\mathbf{xx}}$, besides $\bar{\mathcal{L}}_{\mathbf{uu}}$ and $\bar{\mathcal{L}}_{\mathbf{vv}}$. The result of this observation is that for the discrete time case the positive definiteness of $Q_{\mathbf{uu}}$ and the negative definiteness of $Q_{\mathbf{vv}}$ along the nominal trajectories are not guaranteed. This is not the case with the continuous time formulation of GT-DDP and therefore the continuous version is numerically more stable that the discrete time.

In terms of running time comparison of the two algorithms, continuous time GT-DDP requires the usage of differential equation solvers, while only arithmetic calculations are needed in discrete time GT-DDP. Therefore, the time cost per iteration is faster in the discrete time GT-DDP case, where each iteration is given by an update of the controls.

Examples

In this subsection, we first apply our algorithms on some minimax problem, then we demonstrate how our algorithms can also be utilized to bring some systems to their goal states under stochastic disturbances.



(a) Cost per iteration under discrete GT-DDP.

(b) Cost per iteration under continuous GT-DDP.

Figure 28: Comparison in cost per iteration.

Inverted Pendulum Problem. We first apply our discrete and continuous algorithms on the inverted pendulum with conflicting controls. In particular, the dynamics is given by $I\ddot{\theta} + b\dot{\theta} - mg\ell \sin\theta = \mathbf{u} + \mathbf{v}$, where the parameters are chosen in the simulations as $m = 1\text{kg}$, $\ell = 0.5\text{m}$, $b = 0.1$, $I = m\ell^2$, $g = 9.81\text{kg} \cdot \text{m/sec}^2$. Our goal is to bring the pendulum from the initial state $[\theta, \dot{\theta}] = [\pi, 0]$ to $[\theta, \dot{\theta}] = [0, 0]$.

We cast this problem as a trajectory optimization problem. The cost function is given by

$$J = \int_0^{t_f} (\mathbf{x}^{\mathrm{T}} Q \mathbf{x} + \mathbf{u}^{\mathrm{T}} R_{\mathbf{u}} \mathbf{u} - \mathbf{v}^{\mathrm{T}} R_{\mathbf{v}} \mathbf{v}), \tag{160}$$

where $\mathbf{x} = [\theta, \dot{\theta}]^{\mathrm{T}}$, $Q = \begin{bmatrix} 1, & 0 \\ 0, & 0.1 \end{bmatrix}$, and $R_{\mathbf{u}} = 0.01$, $R_{\mathbf{v}} = 1$.

We set the initial control to be $\mathbf{u} \equiv 0$, $\mathbf{v} \equiv 0$ and the terminal time to be $t_f = 1$. The multiplier $\gamma = 0.8$ in the continuous case. Initial controls and the corresponding initial trajectories of the states are identical in both cases.

We run the algorithms for 30 iterations and the convergence is achieved in both cases. The two algorithms yield the same results in terms of optimal trajectories and corresponding optimal controls, as expected. As can be seen in Figure 28a and 28b, the cost converges in 10 iterations in the discrete case. On the other hand, 4 iterations are needed for the convergence of the cost in the continuous case.

Quadrotor. The dynamic model of the quadrotor includes 16 states: 3 for the position ($\mathbf{r} = (x, y, z)^{\mathrm{T}}$), 3 for the Euler angles ($\Phi = (\phi, \theta, \psi)^{\mathrm{T}}$), 3 for the velocity ($\dot{\mathbf{r}} = (\dot{x}, \dot{y}, \dot{z})^{\mathrm{T}}$), 3 for the body

angular rates ($\dot{\Phi} = (p, q, r)^{\mathrm{T}}$) and 4 for the motor speeds ($\Omega = (\omega_1, \omega_2, \omega_3, \omega_4)^{\mathrm{T}}$). The corresponding dynamics of the quadrotor with conflicting controls is given as follows:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = f(\mathbf{x}) + G(\mathbf{u} - \mathbf{v}), \tag{161}$$

where $\mathbf{x} = [\mathbf{r}, \Phi, \dot{\mathbf{r}}, \dot{\Phi}, \Omega]^{\mathrm{T}} \in \mathbb{R}^{16}$, and $\mathbf{u} = (u_1, u_2, u_3, u_4)^{\mathrm{T}} \in \mathbb{R}^4$ is the stabilizing control vector, where u_1 represents the thrust force, and u_2, u_3, u_4 represent the pitching, rolling, yawing moments, respectively. u_1, u_2, u_3 and u_4 are deviations from the nominal vector $(\omega_h, 0, 0, 0)^{\mathrm{T}}$. $\mathbf{v} \in \mathbb{R}^4$ denotes the destabilizing control. The corresponding cost function is defined as $J = \frac{1}{2}\int_0^{t_f}[(\mathbf{x} - p)^{\mathrm{T}}Q(\mathbf{x} - p) + \mathbf{u}^{\mathrm{T}}R_{\mathbf{u}}\mathbf{u} - \mathbf{v}^{\mathrm{T}}R_{\mathbf{v}}\mathbf{v}] + \frac{1}{2}(\mathbf{x} - p)^{\mathrm{T}}Q_f(\mathbf{x} - p)$, where $p \in \mathbb{R}^{16}$ denotes the desired terminal states. The state dependent running cost is included for better convergence. In the simulation, we set

$$p(i) = \begin{cases} 2\pi, i = 5; \\ 0, \text{otherwise}, \end{cases} \tag{162}$$

and $Q = 0.01Q_f$. $R_{\mathbf{u}} = 0.0001I$ and $R_{\mathbf{v}} = 0.0005I$. $\gamma = 0.5$.

The desired terminal state p is chosen such that the quadrotor would change its pitch angle for 2π and return to its original hovering position. 50 iterations are included to ensure the convergence. The corresponding optimal state trajectories are shown in Figure 29.

3.7 Solution of Stochastic Differential Games using Forward and Backward Stochastic Differential Equations

In this part of the work, we focused our research on designing an efficient algorithm to numerically solve a large class of stochastic differential game problems. The algorithm is a sampling-based scheme which relies on the theory of forward and backward stochastic differential equations (FBSDEs) and their connection to backward PDEs. In particular, we first obtain a probabilistic representation of the solution to the Hamilton Jacobi Isaacs (HJI) PDE, expressed in the form of a system of FBSDEs. This system of FBSDEs is then simulated by employing linear regression techniques. Since the HJI PDE appears in both stochastic differential games and risk-sensitive optimal control problems, the proposed scheme is applicable to both types of stochastic optimal control formulations.

Problem Statement

Let $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}_{t\geq 0}, \mathbb{P})$ be a complete, filtered probability space on which a p-dimensional standard Brownian motion W_t is defined, such that $\{\mathscr{F}_t\}_{t\geq 0}$ is the natural filtration of W_t augmented by all $\mathbb{P}$-null sets. Consider the game-theoretic setting in which the expected game payoff is defined by

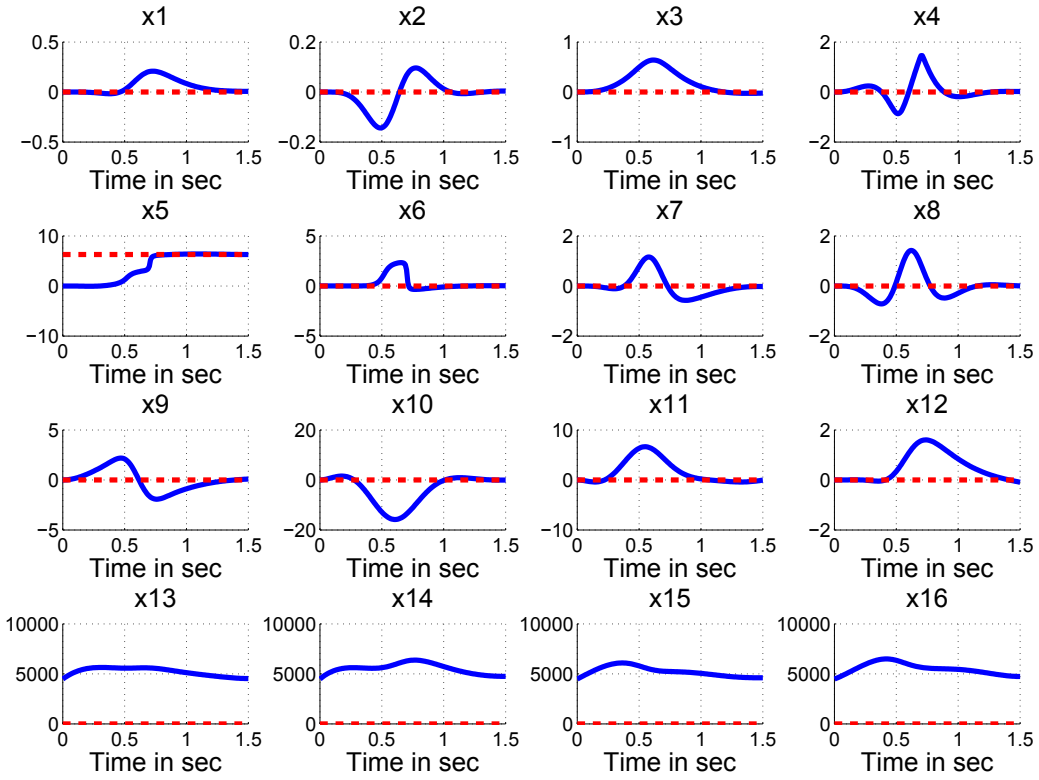


Figure 29: Optimal trajectories of the states in blue. Dashed red lines represent the desired terminal states.

the functional

$$P(\tau, x_\tau; u(\cdot), v(\cdot)) = \mathbb{E}\Big[g(x_T) + \int_\tau^T \big[q(t, x_t) + \tfrac{1}{2} u_t^\top R u_t - \tfrac{1}{2} v_t^\top Q v_t\big] \mathrm{d}t\Big], \tag{163}$$

associated with the stochastic controlled system, which is represented by the Itô stochastic differential equation (SDE)

$$\begin{cases} \mathrm{d}x_t = f(t, x_t)\mathrm{d}t + G(t, x_t)u_t \mathrm{d}t + L(t, x_t)v_t \mathrm{d}t + \Sigma(t, x_t)\mathrm{d}W_t, \quad t \in [\tau, T], \\ x(\tau) = x_\tau, \end{cases} \tag{164}$$

where $T > \tau \geq 0$, T is a fixed time of termination, $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^\nu$ is the minimizing control vector, and $v \in \mathbb{R}^\mu$ is the maximizing control vector. Furthermore, R and Q

are respectively $\nu \times \nu$ and $\mu \times \mu$ positive definite matrices, $g : \mathbb{R}^n \to \mathbb{R}$, $q : [\tau, T] \times \mathbb{R}^n \to \mathbb{R}$, $f : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$, $G : [0, T] \times \mathbb{R}^n \to \mathbb{R}^{n\times\nu}$, $L : [0, T] \times \mathbb{R}^n \to \mathbb{R}^{n\times\mu}$ and $\Sigma : [0, T] \times \mathbb{R}^n \to \mathbb{R}^{n\times p}$ are deterministic functions, that is, they do not depend explicitly on $\omega \in \Omega$. We assume that all standard technical conditions which pertain to the filtered probability space and the regularity of functions are met, in order to guarantee existence, uniqueness of solutions to (164), and a well defined payoff (163). These impose, for example, that the functions g, q, f, G, L and Σ are continuous w.r.t. time t (in case there is explicit dependence), Lipschitz (uniformly in t) with respect to the state variables, and satisfy standard growth conditions over the domain of interest. Furthermore, the square-integrable processes $u : [0, T] \times \Omega \to \mathcal{U} \subseteq \mathbb{R}^\nu$ and $v : [0, T] \times \Omega \to \mathcal{V} \subseteq \mathbb{R}^\mu$ are $\{\mathscr{F}_t\}_{t\geq 0}$-adapted, which essentially translates into the control inputs being non-anticipating, i.e., relying only on past and present information.

The intuitive idea behind the game-theoretic setting is the existence of two players of conflicting interests. The first player controls u and wishes to minimize the payoff P over all choices of v, while the second player wishes to maximize P over all choices of u of his opponent. At any given time, the current state, as well as each opponents' current control action is known to both players. Furthermore, instantaneous switches in both controls are permitted, rendering the problem difficult to solve in general.

The Value Function and the HJI Equation

For any given initial condition (τ, x_τ), we investigate the game of conflicting control actions u, v that minimize (47) under all admissible non-anticipating strategies assigned to $u(\cdot)$, while maximizing it over all admissible non-anticipating strategies assigned to $v(\cdot)$. For the structure imposed on this problem by the form of the cost and dynamics at hand, the Isaacs condition[1] holds, and the payoff is a saddlepoint solution to the following terminal value problem of a second order partial differential equation, known as the Hamilton-Jacobi-Isaacs (HJI) equation, which herein takes the form

$$\begin{cases} V_t + \inf\limits_{u\in\mathcal{U}} \sup\limits_{v\in\mathcal{V}} \left\{ \frac{1}{2}\mathrm{tr}(V_{xx}\Sigma\Sigma^\top) + V_x^\top (f + Gu + Lv) + q + u^\top R u - \frac{1}{2} v^\top Q v \right\} = 0, \qquad (t, x) \in [0, T) \times \mathbb{R}^n, \\ V(T, x) = g(x), \quad x \in \mathbb{R}^n. \end{cases} \tag{165}$$

In the above, function arguments have been suppressed for notational compactness, and V_x and V_{xx} denote the gradient and the Hessian of V, respectively. The term inside the brackets is the Hamiltonian. For the chosen form of the cost integrand, and assuming that the optimal controls lie in the interiors of $\mathcal{U}$ and $\mathcal{V}$, we may carry out the infimum and supremum operations in (165) explicitly by taking the gradient of the Hamiltonian with respect to u and v and setting it equal

[1]The Isaacs condition renders the viscosity solutions of the upper and lower value functions equal, thus making the order of minimization/maximization inconsequential.

to zero, and therefore, for all $(t,x) \in [0,T] \times \mathbb{R}^n$, the optimal controls are given by

$$u^*(t,x) = -R^{-1}G^\top(t,x)V_x(t,x), \tag{166}$$

$$v^*(t,x) = Q^{-1}L^\top(t,x)V_x(t,x). \tag{167}$$

Inserting the above expression back into the original HJI equation and suppressing function arguments for notational brevity, we obtain the equivalent characterization

$$\begin{cases} V_t + \frac{1}{2}\mathrm{tr}(V_{xx}\Sigma\Sigma^\top) + V_x^\top f + q - \frac{1}{2}V_x^\top \Big(GR^{-1}G^\top - LQ^{-1}L^\top \Big) V_x = 0, \quad (t,x) \in [0,T) \times \mathbb{R}^n, \\ V(T,x) = g(x), \quad x \in \mathbb{R}^n. \end{cases} \tag{168}$$

A Feynman-Kac Representation through FBSDEs

There is a close relationship between stochastic differential equations and second-order partial differential equations (PDEs) of parabolic or elliptic type. Specifically, solutions to a certain class of nonlinear PDEs can be represented by solutions to forward-backward stochastic differential equations (FBSDEs), in the same spirit as demonstrated by the well-known Feynman-Kac formulas for linear PDEs. We begin by briefly reviewing FBSDEs.

As a forward process we shall define the square-integrable, $\{\mathscr{F}_s\}_{s\geq 0}$-adapted process $X(\cdot)$[2], which, for any given initial condition $(t,x) \in [0,T] \times \mathbb{R}^n$, satisfies the Itô FSDE

$$\begin{cases} \mathrm{d}X_s = b(s,X_s)\mathrm{d}s + \Sigma(s,X_s)\mathrm{d}W_s, \qquad s \in [t,T], \\ X_t = x. \end{cases} \tag{169}$$

The forward process (169) is also called the *state process* in the literature. We shall denote the solution to the forward SDE (169) as $X_s^{t,x}$, wherein (t,x) are the initial condition parameters.

In contrast to the forward process, the associated backward process is the square-integrable, $\{\mathscr{F}_s\}_{s\geq 0}$-adapted pair $(Y(\cdot), Z(\cdot))$ defined via a BSDE satisfying a terminal condition

$$\begin{cases} \mathrm{d}Y_s = -h(s,X_s,Y_s,Z_s)\mathrm{d}s + Z_s^\top \mathrm{d}W_s \qquad s \in [t,T], \\ Y_T = g(X_T). \end{cases} \tag{170}$$

The function $h(\cdot)$ is called the *generator* or *driver*. The solution is implicitly defined by the initial condition parameters (t,x) of the FSDE since it obeys the terminal condition $g(X_T^{t,x})$. We will similarly use the notation $Y_s^{t,x}$ and $Z_s^{t,x}$ to denote the solution for a particular initial condition parameter (t,x) of the associated FSDE.

[2]While X is a function of s and ω, we shall use X_s for notational brevity.

While FSDEs have a fairly straightforward definition, in the sense that both the SDE and the filtration evolve forward in time, this is not the case for BSDEs. Indeed, since solutions to BSDEs need to satisfy a terminal condition, integration needs to be performed backwards in time in some sense, yet the filtration still evolves forward in time. It turns out that a terminal value problem involving BSDEs admits an adapted (i.e., non-anticipating) solution if we back-propagate the *conditional expectation* of the process, that is, if we set $Y_s \triangleq \mathbb{E}[Y_s|\mathscr{F}_s]$.

Notice that the forward SDE does not depend on Y_s or Z_s. Thus, the resulting system of FBSDEs is said to be *decoupled*. If, in addition, the functions b, Σ, h and g are deterministic, in the sense that they do not depend explicitly on $\omega \in \Omega$, then the adapted solution (Y, Z) exhibits the *Markovian* property; namely, it can be written as deterministic functions of solely time and the state process:

Theorem 4. *(The Markovian Property) – There exist deterministic functions $V(t,x)$ and $d(t,x)$[3] such that the solution $(Y^{t,x}, Z^{t,x})$ of the BSDE (170) is*

$$Y_s^{t,x} = V(s, X_s^{t,x}), \quad Z_s^{t,x} = \Sigma^\top(s, X_s^{t,x}) d(s, X_s^{t,x}), \tag{171}$$

for all $s \in [t,T]$.

We now proceed to state the nonlinear Feynman-Kac type formula, which links the solution of a class of PDEs to that of FBSDEs. Indeed, the following theorem can be proven by an application of Itô's formula:

Theorem 5. *(Nonlinear Feynman-Kac) – Consider the Cauchy problem*

$$\begin{cases} V_t + \frac{1}{2}\mathrm{tr}(V_{xx}\Sigma\Sigma^\top) + V_x^\top b(t,x) + h(t,x,V,\Sigma^\top V_x) = 0, \\ (t,x) \in [0,T) \times \mathbb{R}^n, \quad V(T,x) = g(x), \quad x \in \mathbb{R}^n, \end{cases} \tag{172}$$

wherein the functions Σ, b, h and g satisfy mild regularity conditions. Then (172) admits a unique (viscosity) solution $V : [0,T] \times \mathbb{R}^n \to \mathbb{R}$, which has the following probabilistic representation:

$$V(t,x) = Y_t^{t,x}, \qquad \forall (t,x) \in [0,T] \times \mathbb{R}^n, \tag{173}$$

wherein $(X(\cdot), Y(\cdot), Z(\cdot))$ is the unique adapted solution of the FBSDE system (169)-(170). Furthermore,

$$(Y_s^{t,x}, Z_s^{t,x}) = \Big(V(s, X_s^{t,x}),\ \Sigma^\top(s, X_s^{t,x}) V_x(s, X_s^{t,x}) \Big), \tag{174}$$

for all $s \in [t,T]$, and if (172) admits a classical solution, then (173) provides that classical solution.

[3]By abuse of notation, here (t,x) are symbolic arguments of the functions V and d, and not the initial condition parameters as in $(Y^{t,x}, Z^{t,x})$. Throughout this work, it should be clear from the context whether (t,x) are to be understood as initial condition parameters or symbolic arguments.

A careful comparison between equations (168) and (172) indicates that the nonlinear Feynman-Kac representation can be applied to the HJI equation given by (168) under a certain decomposability condition, stated in the following assumption:

Assumption 2. *There exist matrix-valued functions* $\Gamma : [0,T] \times \mathbb{R}^n \to \mathbb{R}^{p\times\nu}$ *and* $B : [0,T] \times \mathbb{R}^n \to \mathbb{R}^{p\times\mu}$ *such that* $G(t,x) = \Sigma(t,x)\Gamma(t,x)$ *and* $L(t,x) = \Sigma(t,x)B(t,x)$ *for all* $(t,x) \in [0,T] \times \mathbb{R}^n$.

This assumption implies that the range of G and L must be a subset of the range of Σ, and thus excludes the case of a channel containing control input but no noise, although the converse is allowed. Under this assumption, the HJI equation given by (168) becomes

$$\begin{cases} V_t + \frac{1}{2}\mathrm{tr}(V_{xx}\Sigma\Sigma^\top) + V_x^\top f + q - \frac{1}{2}V_x^\top\Sigma\Big(\Gamma R^{-1}\Gamma^\top - BQ^{-1}B^\top\Big)\Sigma^\top V_x = 0, \quad (t,x) \in [0,T) \times \mathbb{R}^n, \\ V(T,x) = g(x), \quad x \in \mathbb{R}^n, \end{cases} \tag{175}$$

in which function arguments have been suppressed, and which satisfies the format of (172) with

$$b(t,x) \equiv f(t,x), \tag{176}$$

and

$$h(t,x,z) \equiv q(t,x) - \tfrac{1}{2}z^\top\big(\Gamma(t,x)R^{-1}\Gamma^\top(t,x) - B(t,x)Q^{-1}B^\top(t,x)\big)z. \tag{177}$$

We may thus obtain the (viscosity) solution of (175) by simulating the system of FBSDE given by (169) and (170). Notice that (169) corresponds to the uncontrolled ($u = 0$, $v = 0$) system dynamics.

Connection to Risk-Sensitive Control

The connection between dynamic games and risk-sensitive stochastic control is well-documented in the literature. Specifically, the optimal controller of a stochastic control problem with exponentiated integral cost (a so-called risk-sensitive problem) turns out to be identical to the minimizing player's unique minimax controller in a stochastic differential game setting. Indeed, consider the problem of minimizing the expected cost given by

$$J(\tau, x_\tau; u(\cdot)) = \epsilon \ln \mathbb{E}\bigg\{ \exp\frac{1}{\epsilon}\bigg[g(x_T) + \int_\tau^T q(t,x_t) + \tfrac{1}{2}u_t^\top R u_t \, \mathrm{d}t\bigg]\bigg\}, \tag{178}$$

where ϵ is a small positive number. The state dynamics are described by the Itô SDE

$$\begin{cases} \mathrm{d}x_t = f(t,x_t)\mathrm{d}t + G(t,x_t)u_t\mathrm{d}t + \sqrt{\dfrac{\epsilon}{2\gamma^2}}\tilde{\Sigma}(t,x_t)\mathrm{d}W_t, \\ t \in [\tau,T], \qquad x(\tau) = x_\tau. \end{cases} \tag{179}$$

Suppressing function arguments for notational compactness, the associated Hamilton-Jacobi-Bellman PDE for this problem is

$$\begin{cases} V_t + \inf\limits_{u\in\mathcal{U}} \left\{ \dfrac{\epsilon}{4\gamma^2}\mathrm{tr}(V_{xx}\tilde{\Sigma}\tilde{\Sigma}^\top) + V_x^\top(f+Gu) + q + u^\top R u + \dfrac{1}{4\gamma^2}V_x^\top\tilde{\Sigma}\tilde{\Sigma}^\top V_x \right\} = 0, \quad (t,x)\in[0,T)\times\mathbb{R}^n, \\ V(T,x) = g(x), \quad x\in\mathbb{R}^n. \end{cases} \tag{180}$$

The infimum operation can be performed explicitly, and yields the optimal control $u^*(t,x) = -R^{-1}G^\top(t,x)V_x(t,x)$. Setting $\Sigma = \sqrt{\epsilon/2\gamma^2}\tilde{\Sigma}$ and substituting the optimal control in the PDE (180) we readily obtain the equivalent characterization

$$\begin{cases} V_t + \frac{1}{2}\mathrm{tr}(V_{xx}\Sigma\Sigma^\top) + V_x^\top f + q - \frac{1}{2}V_x^\top\left(GR^{-1}G^\top - \dfrac{1}{\epsilon}\Sigma\Sigma^\top\right)V_x = 0, \qquad (t,x)\in[0,T)\times\mathbb{R}^n, \\ V(T,x) = g(x), \quad x\in\mathbb{R}^n. \end{cases} \tag{181}$$

The above equation is merely a special case of equation (168) obtained for the game-theoretic version, if one substitutes $Q = (1/\epsilon)I$ and $L = \Sigma$. Notice that this special case of L automatically satisfies Assumption 1 with B being the identity matrix. Thus, imposing the same decomposability condition on G, the solution to the risk-sensitive stochastic optimal control problem can be obtained by simulating the system of FBSDEs given by (169) and (170) using the definitions (176) and (177).

Approximating the Solution of FBSDEs

The solution of FBSDEs has been studied to a great extent independently from its connection to PDEs, mainly within the field of mathematical finance. Though several generic schemes exist, in this work we employed a modification which exploits the regularity present in FBSDEs that arise from the application of the nonlinear Feynman-Kac lemma.

We begin by selecting a time grid $\{t = t_0 < \ldots < t_N = T\}$ for the interval $[t,T]$, and denote by $\Delta t_i \triangleq t_{i+1} - t_i$ the $(i+1)$-th interval of the grid (which can be selected to be constant) and $\Delta W_i \triangleq W_{t_{i+1}} - W_{t_i}$ the $(i+1)$-th Brownian motion increment. For notational brevity, we also denote $X_i \triangleq X_{t_i}$. The simplest discretized scheme for the forward process is the Euler scheme, which is also called *Euler-Maruyama* scheme:

$$\begin{cases} X_{i+1} \approx X_i + b(t_i, X_i)\Delta t_i + \Sigma(t_i, X_i)\Delta W_i, \\ i = 1,\ldots,N, \qquad X_0 = x. \end{cases} \tag{182}$$

Several alternative, higher order schemes exist that can be selected in lieu of the Euler scheme. To discretize the backward process, we further introduce the notation $Y_i \triangleq Y_{t_i}$ and $Z_i \triangleq Z_{t_i}$. Then, recalling that adapted BSDE solutions impose $Y_s \triangleq \mathbb{E}[Y_s|\mathscr{F}_s]$ and $Z_s \triangleq \mathbb{E}[Z_s|\mathscr{F}_s]$ (i.e., a

back-propagation of the conditional expectations), we approximate equation (170) by

$$Y_i = \mathbb{E}[Y_i|\mathscr{F}_{t_i}] \approx \mathbb{E}[Y_{i+1} + h(t_{i+1}, X_{i+1}, Y_{i+1}, Z_{i+1})\Delta t_i|X_i]. \tag{183}$$

Notice that in the last equality the term $Z_i^\top \Delta W_i$ in (170) vanishes because of the conditional expectation (ΔW_i is zero mean), and we replace $\mathscr{F}_{t_i}$ with X_i in light of the Markovian property presented earlier. By virtue of equation (174), the Z-process in (170) corresponds to the term $\Sigma^\top(s, X_s^{t,x})v_x(s, X_s^{t,x})$. Therefore we can write

$$\begin{aligned} Z_i &= \mathbb{E}[Z_i|\mathscr{F}_{t_i}] = \mathbb{E}[\Sigma^\top(t_i, X_i)\nabla_x v(t_i, X_i)|X_i] \\ &= \Sigma^\top(t_i, X_i)\nabla_x v(t_i, X_i), \end{aligned} \tag{184}$$

which naturally requires knowledge of the solution at time t_i on a neighborhood x, $v(t_i, x)$. The backpropagation is initialized at

$$Y_T = g(X_T), \qquad Z_T = \Sigma(T, X_T)^\top \nabla_x g(X_T), \tag{185}$$

for a $g(\cdot)$ which is differentiable almost everywhere. There are several ways to approximate the conditional expectation in (183), however in this work we shall employ the Least Squares Monte Carlo (LSMC) method, which we shall briefly review in what follows.

The LSMC method addresses the general problem of numerically estimating conditional expectations of the form $\mathbb{E}[Y|X]$ for square integrable random variables X and Y, if one is able to sample M independent copies of pairs (X, Y). The method itself is based on the principle that the conditional expectation of a random variable can be modeled as a function of the variable on which it is conditioned on, that is, $\mathbb{E}[Y|X] = \phi^*(X)$, where ϕ^* solves the infinite dimensional minimization problem

$$\phi^* = \arg\min_{\phi} \mathbb{E}[|\phi(X) - Y|^2], \tag{186}$$

and ϕ ranges over all measurable functions with $\mathbb{E}[|\phi(X)|^2] < \infty$. A finite-dimensional approximation of this problem can be obtained if one decomposes $\phi(\cdot) \approx \sum_{i=1}^k \varphi_i(\cdot)\alpha_i = \varphi(\cdot)\alpha$, with $\varphi(\cdot)$ being a row vector of k predetermined basis functions and α a column vector of constants, thus solving $\alpha^* = \arg\min_{\alpha\in\mathbb{R}^k} \mathbb{E}[|\varphi(X)\alpha - Y|^2]$, with k being the dimension of the basis. Finally, this problem can be simplified to a linear-least squares problem if one substitutes the expectation operator with its empirical estimator, thus obtaining

$$\alpha^* = \arg\min_{\alpha\in\mathbb{R}^k} \frac{1}{M}\sum_{j=1}^{M} |\varphi(X^j)\alpha - Y^j|^2, \tag{187}$$

wherein (X^j, Y^j), $j = 1, \ldots, M$ are independent copies of (X, Y). Introducing the notation

$$\Phi(X) = \begin{bmatrix} \varphi(X^1) \\ \vdots \\ \varphi(X^M) \end{bmatrix} \in \mathbb{R}^{M\times k}, \tag{188}$$

the solution to this least-squares problem can be obtained by directly solving the normal equation, i.e.,

$$a^* = \left(\Phi^\top(X)\Phi(X)\right)^{-1}\Phi^\top(X)\left(\begin{bmatrix} Y^1 \\ \vdots \\ Y^M \end{bmatrix}\right), \tag{189}$$

or by performing gradient descent. The LSMC estimator for the conditional expectation assumes then the form $\mathbb{E}[Y|X=x] = \phi^*(x) \approx \varphi(x)a^*$.

Returning to our problem, we may apply the LSMC method to approximate the conditional expectation in equation (183) for each time step. To this end, we require a vector of basis functions φ for the approximation of $\mathbb{E}[Y_i|X_i]$. Although the basis functions can be different at each time step, we shall use the same symbol for notational simplicity. Then, Monte Carlo simulation is performed by sampling M independent trajectories $\{X_i^m\}_{i=1,\dots,N}$, in which the index $m = 1,\dots,M$ specifies a particular Monte Carlo trajectory. Whenever this index is not present, the entirety with respect to this index is to be understood. The numerical scheme is initialized at the terminal time T and is iterated backwards along the entire time grid, until the starting time instant has been reached. At each time step t_i, we are given M pairs of data (Y_i^m, X_i^m)[4] on which we perform linear regression to estimate the conditional expectation of Y_i as a function of x at the time step t_i. This provides us an approximation of the Value function v at time t_i for the neighborhood of the state space that has been explored by the sample trajectories at that time instant, since $v(t_i,x) = \mathbb{E}[Y_i|X_i=x] \approx \varphi(x)\alpha_i$. We then replace $Y_i^m = \mathbb{E}[Y_i^m|X_i^m] \approx \varphi(X_i^m)\alpha_i$, thereby treating the conditional expectation as a projection operator. Finally, the approximation of the conditional expectation of Z_i is obtained by taking the gradient with respect to x on $v(t_i,x)$, evaluating it at X_i^m, and scaling it with Σ

$$Z_i^m \approx \Sigma(t_i, X_i^m)^\top \nabla_x \varphi(X_i^m)\alpha_i. \tag{190}$$

Concluding one iteration, this process is repeated for $t_{i-1},\dots,t_1$. Note that this approach requires the basis functions $\varphi(\cdot)$ of our choice to be differentiable almost everywhere, so that $\nabla_x\varphi(x)$ is available in analytical form for almost any x. The proposed algorithm is then summarized as

$$\begin{cases} \text{Initialize} : Y_T = g(X_T), \quad Z_T = \Sigma(T,X_T)^\top \nabla_x g(X_T), \\ \alpha_i = \arg\min\limits_{\alpha} \dfrac{1}{M}\left\|\Phi(X_i)\alpha - \left(Y_{i+1} + \Delta t_i h(t_{i+1}, X_{i+1}, Y_{i+1}, Z_{i+1})\right)\right\|^2, \\ Y_i = \Phi(X_i)\alpha_i, \quad Z_i^m = \Sigma(t_i, X_i^m)^\top \nabla_x \varphi(X_i^m)\alpha_i, \end{cases} \tag{191}$$

where $m = 1,\dots,M$ and the matrix Φ defined in (188). Again, the minimizer in (191) can be obtained by directly solving the normal equation, i.e.,

$$a_i = \left(\Phi^\top(X_i)\Phi(X_i)\right)^{-1}\Phi^\top(X_i)\left(Y_{i+1} + \Delta t_i h(t_{i+1}, X_{i+1}, Y_{i+1}, Z_{i+1})\right), \tag{192}$$

[4]Here, Y_i^m denotes the quantity $Y_{i+1}^m + \Delta t_i h(t_{i+1}, X_{i+1}^m, Y_{i+1}^m, Z_{i+1}^m)$, which is the Y_i^m sample value before the conditional expectation operator has been applied.

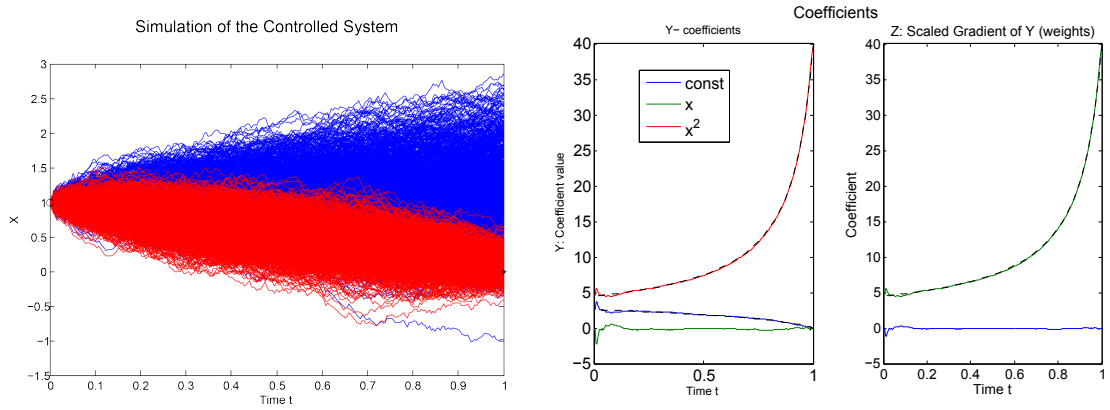
or by performing gradient descent. The essential algorithm output is the collection of a_i's, that is, the basis function coefficients at each time instant, which are needed to recover the Value function approximation for the particular area of the state space that is explored by the forward process.

Simulation Results

To evaluate the algorithm's performance, two simulations were performed on scalar systems, for which, owing to their simplicity, we have the opportunity to evaluate the system behavior.

A Linear System Example: The first example used is a scalar linear system for which the analytic solution can be recovered. Specifically, for a very high maximizer control weight Q, we expect the solution to be almost identical to the LQR solution, which is available in closed form. We simulate the algorithm for $\mathrm{d}x = (0.2x + u + 0.5xv)\mathrm{d}t + 0.5\mathrm{d}w$, with $q(t,x) = 0$, $R = 2$, $x(0) = 1$, $T = 1$ and $g(x_T) = 40x_T^2$, thus penalizing deviation from the origin at the time of termination, T. For Q, the maximizing control cost factor, we selected varying values ranging from 5 to 50,000. In the latter case, we expect to recover the LQR coefficients. For the purposes of comparison with the closed form solution, the set of basis functions for Y was selected to be $[1\ x\ x^2]^\top$. For the LQR controller, the coefficients correspond to the basis functions $[1\ x^2]^\top$. Two thousand trajectories were generated on a time grid of $\Delta t = 0.004$. Fig. 30 shows that, indeed, for very hight values of Q the algorithm recovers the correct theoretic LQR coefficients, while Fig. 31 depicts simulations for the case in which the maximizing control is allowed to act on the system when it is relatively cheap. In this case, we can see that because the maximizer has enough control authority, the equilibrium has moved away from the desired value of $x(T) = 0$, as expected.

A Nonlinear System Example: To demonstrate that the scheme can accommodate nonlinearity in the dynamics, we applied the algorithm on the system $\mathrm{d}x = (4\cos x + u + 0.5xv)\mathrm{d}t + 0.5\mathrm{d}w$. The drift was replaced by a nonlinear term to introduce an additional behavior to the open-loop system trajectories. The results are depicted in Fig. 32. From the shape of the value function in Fig. 32(b) it is seen that the value is relatively flat at the beginning since there is no state-dependent running cost and becomes progressively quadratic at the final time owing to the boundary condition $V(T, x_T) = 40x_T^2$. Note, however, that Fig. 32(b) shows the value function over a rectangular grid. In fact, we have an accurate estimate of the value function only over the area of the state space visited by the sampled (open-loop) trajectories. In that sense, the areas not visited by the system are extrapolated based on the basis functions chosen to represent V.



(a) (b)

Figure 30: Simulation of the system with very high maximizing control cost weight $Q = 50,000$. (a) Controlled trajectories (red) vs. uncontrolled (blue), (b) Y and Z coefficients, compared to those obtained by the closed form solution of the LQR if the maximizing control was not present (black dashed lines). We observe that for a high maximizing control cost, the obtained coefficients match those of the LQR as expected.

4 List of Publications

1. J.P. de la Croix and M. Egerstedt. Analyzing Human-Swarm Interactions Using Control Lyapunov Functions and Optimal Control. *Networks and Heterogeneous Media*, Vol. 10, No. 3, pp. 609-630, Sept. 2015.

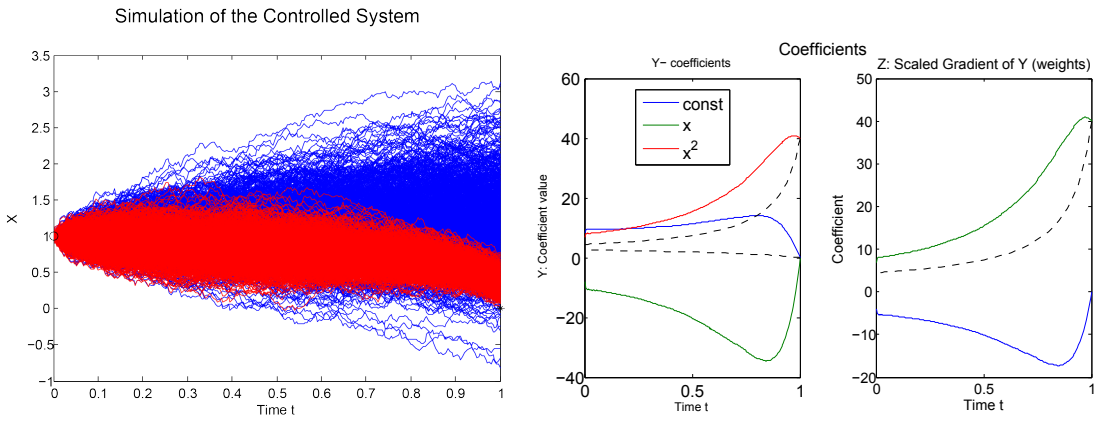
2. T. Setter, A. Fouraker, M. Egerstedt, and H. Kawashima. Haptic Interactions with Multi-Robot Swarms Using Manipulability. *Journal of Human-Robot Interaction*, Vol. 4, No. 1, pp. 60-74, 2015.

3. S. Lee, Y. Diaz-Mercado, and M. Egerstedt. Multi-Robot Control Using Time-Varying Density Functions. *IEEE Transactions on Robotics*, Vol. 31, No. 2, pp. 489-493, Apr. 2015.

4. H. Kawashima and M. Egerstedt. Manipulability of Leader-Follower Networks Under a Rigid-Link Approximation. *Automatica*, Vol. 50, No. 3, pp. 695-796, March 2014.

5. Y. Diaz-Mercado, S.G. Lee, and M. Egerstedt. Human-Swarm Interactions via Coverage of Time-Varying Densities. *Control of Human-Robot Collaboration*, Y. Wang and F. Zhang (Eds.), Springer-Verlag. To appear 2017.

6. M. Egerstedt, J.P. de la Croix, H. Kawashima, and P. Kingston. Interacting with Networks of Mobile Agents. *Large-Scale Networks in Engineering and Life Sciences*, P. Brenner, R.



(a) (b)

Figure 31: Simulation of the system with small maximizing control cost weight $Q = 5$. (a) Controlled trajectories (red) vs. uncontrolled (blue), (b) Y and Z coefficients, compared to those obtained by the closed form solution of the LQR if the maximizing control was not present (black dashed lines).

Findeisen, D. Flockerzi, U. Reichl, and K. Sundmacher (Eds.), Birkhauser, pp. 199-224, 2015.

7. J.P. de la Croix and M. Egerstedt. A Control Lyapunov Function Approach to Human-Swarm Interactions. *American Control Conference*, Chicago, IL, July 2015.

8. T. Setter, H. Kawashima, and M. Egerstedt. Team-Level Manipulability Properties for Human-Swarm Interactions. *American Control Conference*, Chicago, IL, July 2015.

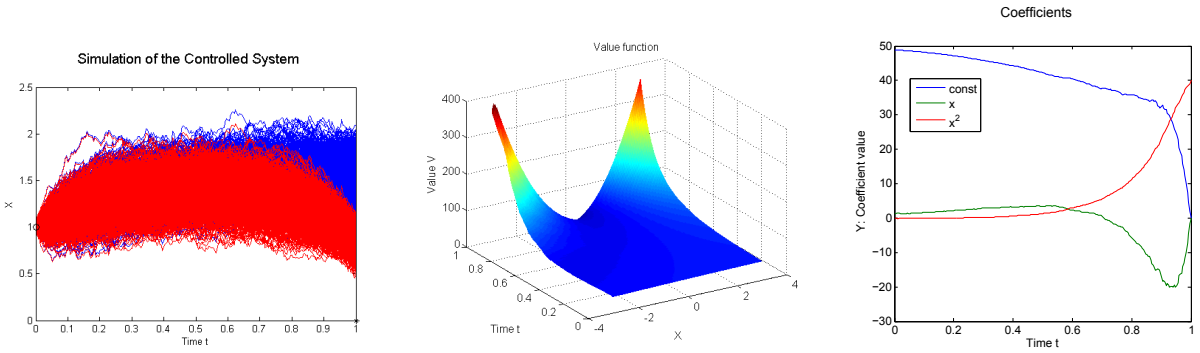
9. Y. Diaz-Mercado, S. Lee, and M. Egerstedt. Distributed Dynamic Density Coverage for Human-Swarm Interactions. *American Control Conference*, Chicago, IL, July 2015.

10. Bakolas, E. and Tsiotras, P., "Optimal Synthesis of the Zermelo–Markov–Dubins Problem in a Constant Drift Field," *Journal of Optimization Theory and Applications*, Vol. 156, No. 2, pp. 469–492, 2013, doi: 10.1007/s10957-012-0128-0

11. Bakolas, E. and Tsiotras, P., "Optimal Partitioning for Spatiotemporal Coverage in a Drift Field," *Automatica*, Vol. 49, No. 7, pp. 2064–2073, July 2013, doi: 10.1016/j.automatica.2013.04.013

12. Anderson, R., Bakolas, E., Milutinović, D., and Tsiotras, P., "Optimal Feedback Guidance of a Small Aerial Vehicle in a Stochastic Wind," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, pp. 975–985, July-August 2013, doi: 10.2514/1.59512

13. Sun, W., and Tsiotras, P., "A Sequential Pursuer-Target Assignment Problem Under External Disturbances," *52nd IEEE Conference on Decision and Control Conference*, Florence, Italy, Dec. 10–13, 2013, pp. 3994–3999, doi:10.1109/CDC.2013.6760500



(a) (b) (c)

Figure 32: Simulation of the nonlinear system with small maximizing control cost weight $Q = 5$. (a) Controlled trajectories (red) vs. uncontrolled (blue), (b) The Value function, (c) Y coefficients.

14. Exarchos, I., and Tsiotras, P., "An Asymmetric Version of the Two Car Pursuit-Evasion Game," *53rd IEEE Conference on Decision and Control*, Los Angeles, CA, Dec. 15–17, 2014, pp. 4272–4277, doi:10.1109/CDC.2014.7040055

15. Exarchos, I., Tsiotras, P., and Pachter, M.,"On the Suicidal Pedestrian Differential Game ," *Journal of Dynamic Games and Applications*, Springer 2014, doi: 10.1007/s13235-014-0130-2.

16. Sun, W. and Tsiotras, P., "An Optimal Evader Strategy in a Two-Pursuer One-Evader Problem," *53rd IEEE Conference on Decision and Control*, Los Angeles, CA, Dec. 15–17, 2014, pp. 4266-4271, doi:10.1109/CDC.2014.7040054

17. Exarchos, I., Tsiotras, P., and Pachter, M., "UAV Collision Avoidance based on the Solution of the Suicidal Pedestrian Differential Game," *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, January 4–8, 2016.

18. Exarchos, I., Theodorou, E., and Tsiotras, P., "Game-Theoretic and Risk-Sensitive Stochastic Optimal Control via Forward and Backward Stochastic Differential Equations," *55th IEEE Conference on Decision and Control*, Las Vegas, NV, December 12–14, 2016.

19. Sun, W., and Tsiotras, P., "Pursuit Evasion Game of Two Players under an External Flow Field," *American Control Conference*, Chicago, IL, July 1–3, 2015, pp. 5617–5622, doi:10.1109/ACC.2015.7172219

20. Sun, W., Theodorou, E. and Tsiotras, P., "Game Theoretic Continuous Time Differential Dynamic Programming," *American Control Conference*, Chicago, IL, July 1–3, 2015, pp. 5593–5598, doi:10.1109/ACC.2015.7172215

21. Sun, W., Tsiotras, P., Lolla, T., Subramani, D. N., and Lermusiaux, P. F. J., "Pursuit-Evasion Games in Dynamic Flow Fields via Reachability Set Analysis," *American Control Conference*, Seattle, WA, May 24-26, 2017 (submitted).

22. Sun, W., Tsiotras, P., Lolla, T., Subramani, D., and Lermusiaux, P., "Multiple-Pursuer-One-Evader Pursuit Evasion Game in Dynamic Flow Fields," *AIAA Journal of Guidance, Control, and Dynamics* (submitted).

23. Sun, W., Theodorou, E., and Tsiotras, P., "Stochastic Game Theoretic Trajectory Optimization In Continuous Time," *55th IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 12–14, 2016.

24. Sun, W., Pan, Y., Theodorou, E., and Tsiotras, P., "Min-Max Differential Dynamic Programming: Continuous and Discrete Time Formulations," *International Journal of Control*, (in preparation).

AFOSR Deliverables Submission Survey

Response ID:7074 Data

1.

Report Type

Final Report

Primary Contact Email

Contact email if there is a problem with the report.

magnus@gatech.edu

Primary Contact Phone Number

Contact phone number if there is a problem with the report

404 894 3484

Organization / Institution name

Georgia Institute of Technology

Grant/Contract Title

The full title of the funded effort.

Motion Coordination and Adaptation Using Deception and Human Interactions

Grant/Contract Number

AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-13-1-0029

Principal Investigator Name

The full name of the principal investigator on the grant or contract.

Magnus Egerstedt

Program Officer

The AFOSR Program Officer currently assigned to the award

Frederick Leve

Reporting Period Start Date

01/01/2013

Reporting Period End Date

06/30/2016

Abstract

This project developed fundamental, new tools and techniques for how to
structure the coordination and control strategies in teams of mobile robots. In particular, two general thrust areas were pursued, focusing on human-swarm interactions and pursuit-evasion-based motion control strategies. Although interesting in their own rights, the unifying theme behind these two different thrusts is the notion of intent, where the first thrust, which can be thought of as evolving at a higher level of abstraction, focused on how user intent can be injected into a network of mobile agents in a fundamentally sound manner. The second thrust, in turn, focused on how the intent can be hidden in order to produce effective, deception-based coordination and pursuit strategies.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

DISTRIBUTION A: Distribution approved for public release.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

sf0298.pdf

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

AFOSR-FINAL-REPORT.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

J.P. de la Croix and M. Egerstedt. Analyzing Human-Swarm Interactions Using Control Lyapunov Functions and Optimal Control. Networks and Heterogeneous Media, Vol. 10, No. 3, pp. 609-630, Sept. 2015.

T. Setter, A. Fouraker, M. Egerstedt, and H. Kawashima. Haptic Interactions with Multi-Robot Swarms Using Manipulability. Journal of Human-Robot Interaction, Vol. 4, No. 1, pp. 60-74, 2015.

S. Lee, Y. Diaz-Mercado, and M. Egerstedt. Multi-Robot Control Using Time-Varying Density Functions. IEEE Transactions on Robotics, Vol. 31, No. 2, pp. 489-493, Apr. 2015.

H. Kawashima and M. Egerstedt. Manipulability of Leader-Follower Networks Under a Rigid-Link Approximation. Automatica, Vol. 50, No. 3, pp. 695-796, March 2014.

Y. Diaz-Mercado, S.G. Lee, and M. Egerstedt. Human-Swarm Interactions via Coverage of Time-Varying Densities. Control of Human-Robot Collaboration, Y. Wang and F. Zhang (Eds.), Springer-Verlag. To appear 2017.

M. Egerstedt, J.P. de la Croix, H. Kawashima, and P. Kingston. Interacting with Networks of Mobile Agents. Large-Scale Networks in Engineering and Life Sciences, P. Brenner, R. Findeisen, D. Flockerzi, U. Reichl, and K. Sundmacher (Eds.), Birkhauser, pp. 199-224, 2015.

J.P. de la Croix and M. Egerstedt. A Control Lyapunov Function Approach to Human-Swarm Interactions. American Control Conference, Chicago, IL, July 2015.

T. Setter, H. Kawashima, and M. Egerstedt. Team-Level Manipulability Properties for Human-Swarm Interactions. American Control Conference, Chicago, IL, July 2015.

Y. Diaz-Mercado, S. Lee, and M. Egerstedt. Distributed Dynamic Density Coverage for Human-Swarm Interactions. American Control Conference, Chicago, IL, July 2015.

Bakolas, E. and Tsiotras, P., Optimal Synthesis of the Zermelo-Markov-Dubins Problem in a Constant Drift Field," Journal of Optimization Theory and Applications, Vol. 156, No. 2, pp. 469-492, 2013, doi: 10.1007/s10957-012-0128-0

Bakolas, E. and Tsiotras, P., Optimal Partitioning for Spatiotemporal Coverage in a Drift Field," Automatica, Vol. 49, No. 7, pp. 2064-2073, July 2013, doi: 10.1016/j.automatica.2013.04.013

Anderson, R., Bakolas, E., Milutinovic, D., and Tsiotras, P., Optimal Feedback Guidance of a Small Aerial Vehicle in a Stochastic Wind," AIAA Journal of Guidance, Control, and Dynamics, Vol. 36, No. 4, pp. 975-985, July-August 2013, doi: 10.2514/1.59512

Sun, W., and Tsiotras, P., A Sequential Pursuer-Target Assignment Problem Under External Disturbances," 52nd IEEE Conference on Decision and Control Conference, Florence, Italy, Dec. 10-13, 2013, pp. 3994-

3999, doi:10.1109/CDC.2013.6760500

Exarchos, I., and Tsiotras, P., An Asymmetric Version of the Two Car Pursuit-Evasion Game," 53rd IEEE Conference on Decision and Control, Los Angeles, CA, Dec. 15-17, 2014, pp. 4272-4277, doi:10.1109/CDC.2014.7040055

Exarchos, I., Tsiotras, P., and Pachter, M.,On the Suicidal Pedestrian Differential Game ," Journal of Dynamic Games and Applications, Springer 2014, doi: 10.1007/s13235-014-0130-2.

Sun, W. and Tsiotras, P., An Optimal Evader Strategy in a Two-Pursuer One-Evader Problem," 53rd IEEE Conference on Decision and Control, Los Angeles, CA, Dec. 15-17, 2014, pp. 4266-4271, doi:10.1109/CDC.2014.7040054

Exarchos, I., Tsiotras, P., and Pachter, M., UAV Collision Avoidance based on the Solution of the Suicidal Pedestrian Differential Game," AIAA Guidance, Navigation, and Control Conference, San Diego, CA, January 4-8, 2016.

Exarchos, I., Theodorou, E., and Tsiotras, P., Game-Theoretic and Risk-Sensitive Stochastic Optimal Control via Forward and Backward Stochastic Differential Equations," 55th IEEE Conference on Decision and Control, Las Vegas, NV, December 12-14, 2016.

Sun, W., and Tsiotras, P., Pursuit Evasion Game of Two Players under an External Flow Field," American Control Conference, Chicago, IL, July 1-3, 2015, pp. 5617-5622, doi:10.1109/ACC.2015.7172219

Sun, W., Theodorou, E. and Tsiotras, P., Game Theoretic Continuous Time Differential Dynamic Programming," American Control Conference, Chicago, IL, July 1-3, 2015, pp. 5593-5598, doi:10.1109/ACC.2015.7172215

Sun, W., Tsiotras, P., Lolla, T., Subramani, D. N., and Lermusiaux, P. F. J., Pursuit-Evasion Games in Dynamic Flow Fields via Reachability Set Analysis," American Control Conference, Seattle, WA, May 24-26, 2017 (submitted).

Sun, W., Tsiotras, P., Lolla, T., Subramani, D., and Lermusiaux, P., Multiple-Pursuer-One-Evader Pursuit Evasion Game in Dynamic Flow Fields," AIAA Journal of Guidance, Control, and Dynamics (submitted).

Sun, W., Theodorou, E., and Tsiotras, P., Stochastic Game Theoretic Trajectory Optimization In Continuous Time," 55th IEEE Conference on Decision and Control, Las Vegas, NV, Dec. 12-14, 2016.

New discoveries, inventions, or patent disclosures:

Do you have any discoveries, inventions, or patent disclosures to report for this period?

No

Please describe and include any notable dates

Do you plan to pursue a claim for personal or organizational intellectual property?

Changes in research objectives (if any):

N/A

Change in AFOSR Program Officer, if any:

This program was initiated under Fariba Fahroo and was taken over by Fred Leve.

Extensions granted or milestones slipped, if any:

N/A

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Oct 06, 2016 17:26:30 Success: Email Sent to: magnus@gatech.edu